Distributed Matching Scheme and a Flexible Deterministic Matching Algorithm for Arbitrary Systems

IPAC 2016, Busan, Korea

Yu-Chiu Chao

May 9th, 2016

Work Supported in Part by National Science & Engineering Research Council of Canada

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Exploring Alternative Approaches to Matching

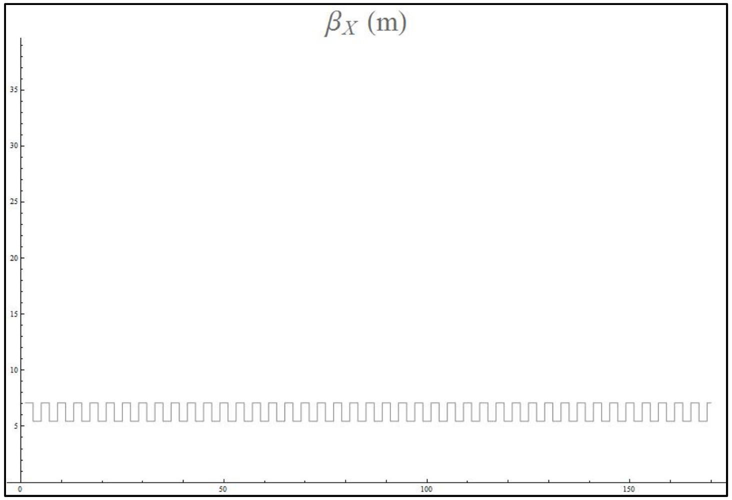
- Performance Improvements through Distributed Matching
  - Envelope and Jitter Control Not Limited by Geographical Location
  - Avoiding Beam Blowup & Optical Sensitivity Due to Drastic Matching
  - Improving Error Tolerance & Dynamic Correction Capability
- A New Approach to Matching Algorithm
  - Robustness and Determinism
  - Logic and Insight
  - Flexibility and Control
  - Solution Capability – Less Vulnerable to Optics/System Complexity
- Advantages through Operational Implementation
  - Pre-Computed Matching Solutions
  - Speed – Major Computation Done Offline
  - User Control and Options

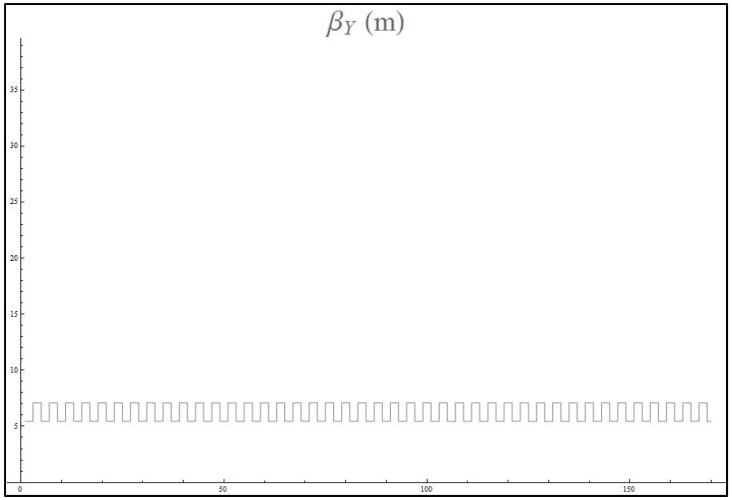
Motivation for Distributed Matching

Motivation for Distributed Matching

30° FODO Lattice

Design $\beta_{X/Y}$



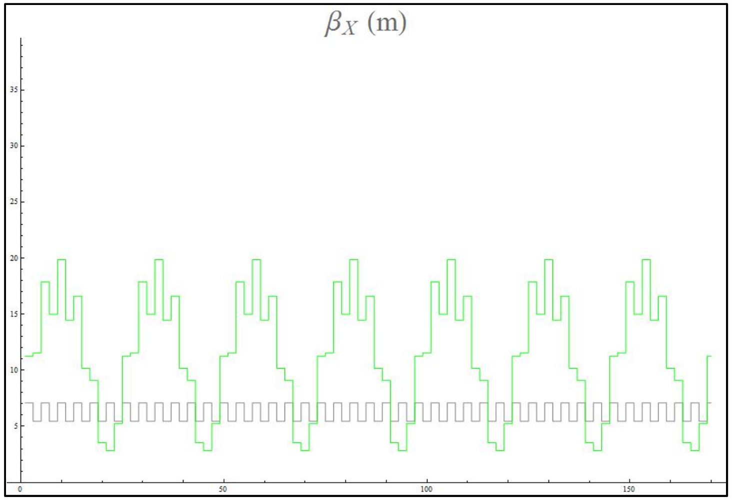


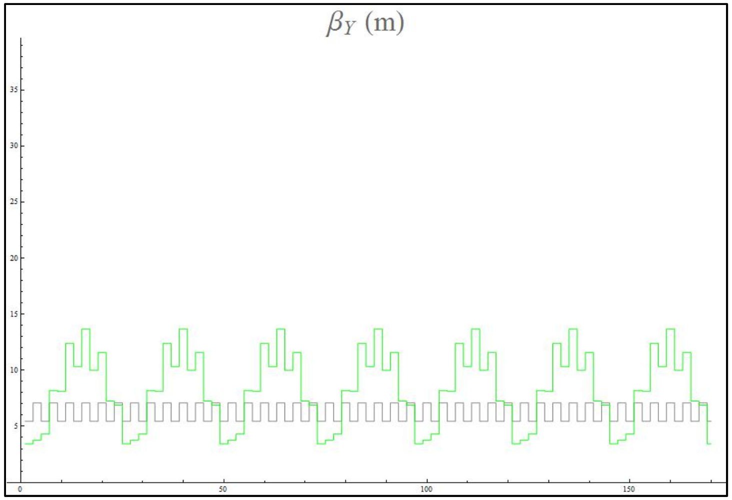
Motivation for Distributed Matching

30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched





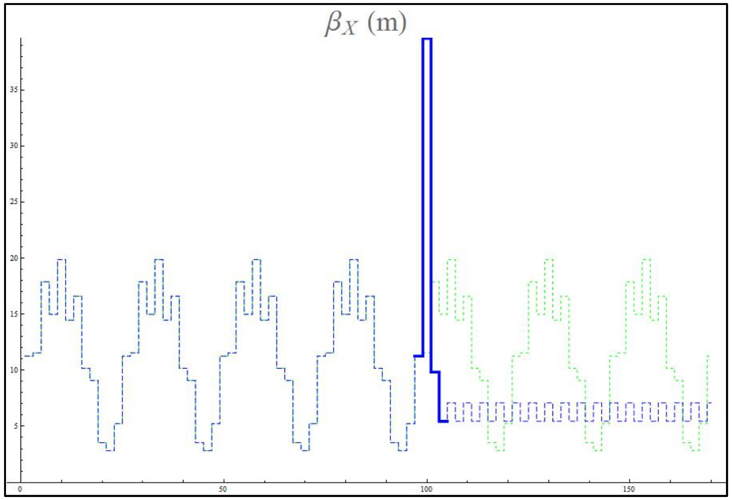
Motivation for Distributed Matching

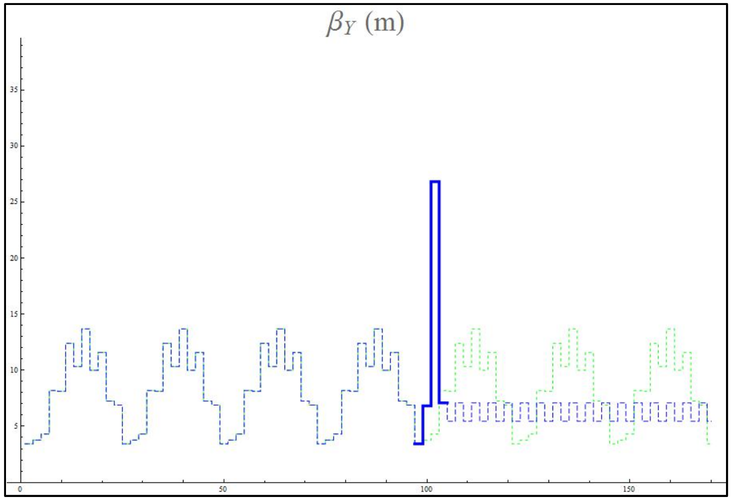
30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching





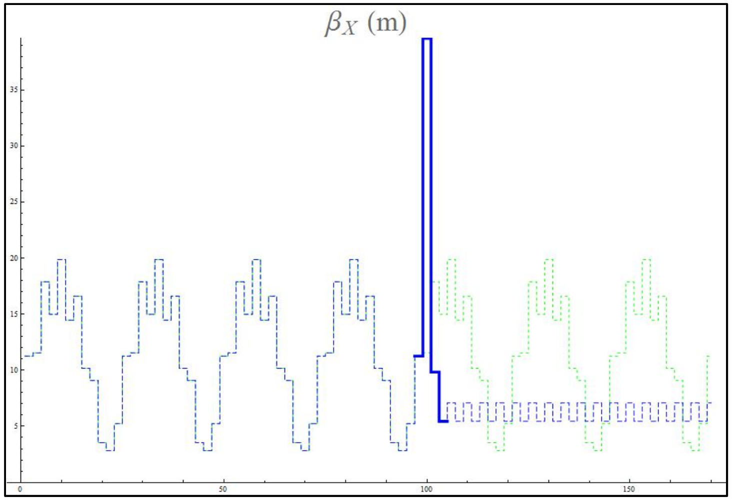
Motivation for Distributed Matching

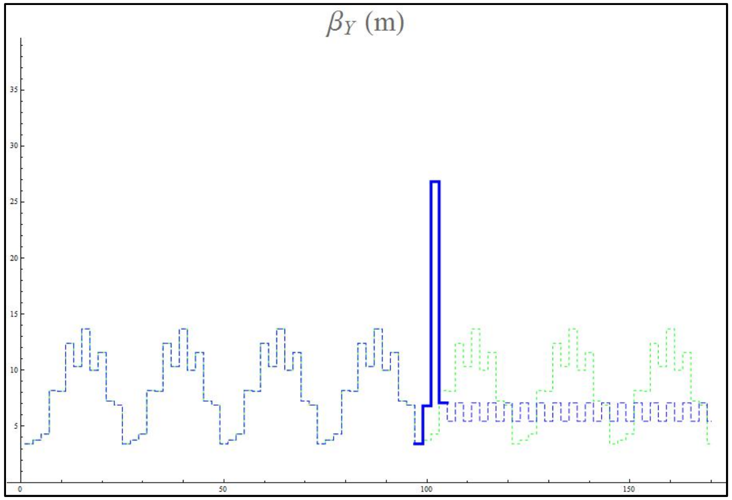
30° FODO Lattice

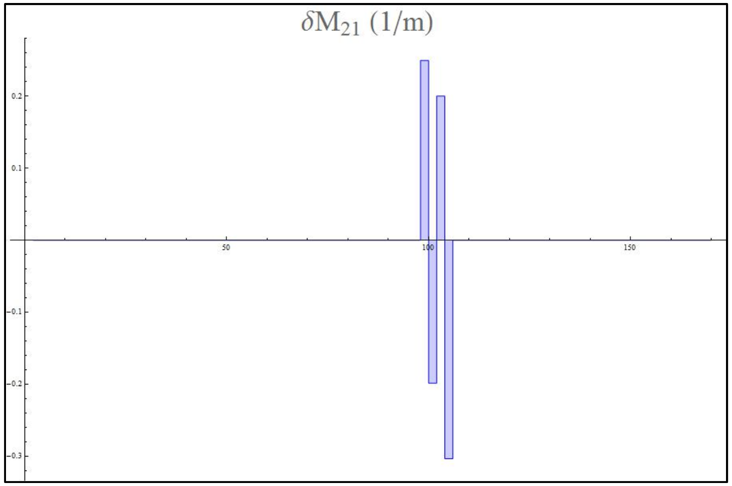
Design $\beta_{X/Y}$

Mismatched

Local Matching







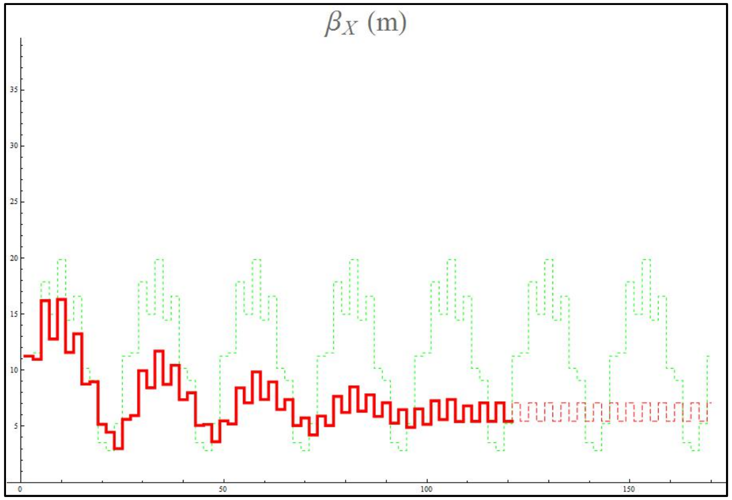
Motivation for Distributed Matching

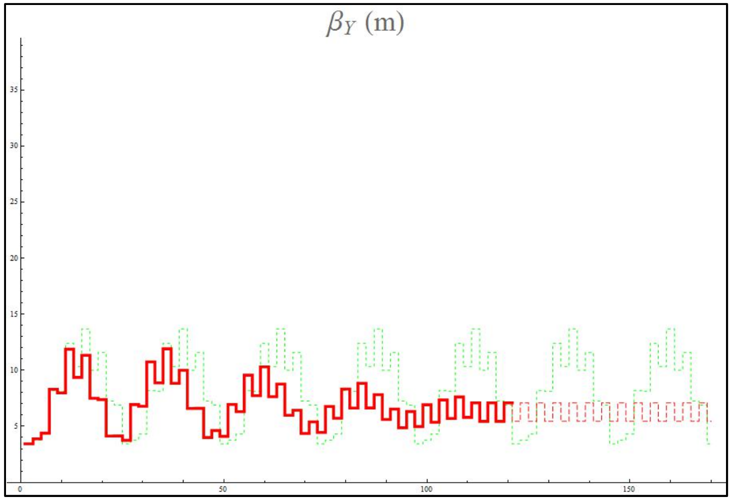
30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching





Distributed Matching over 7.5 cells at source

Motivation for Distributed Matching

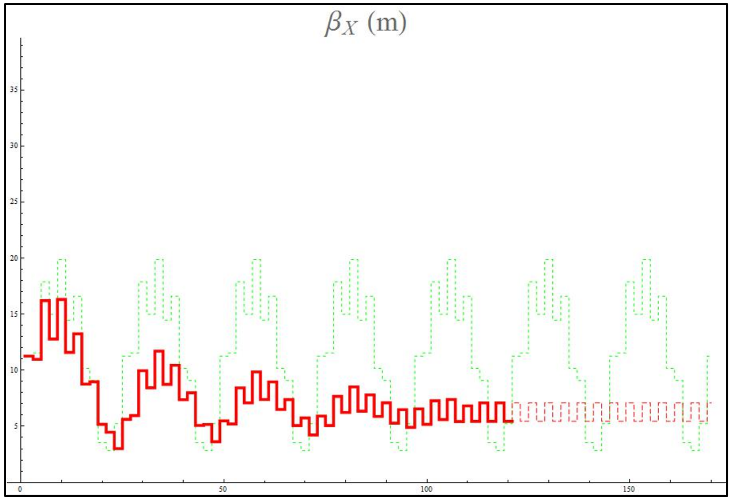
30° FODO Lattice

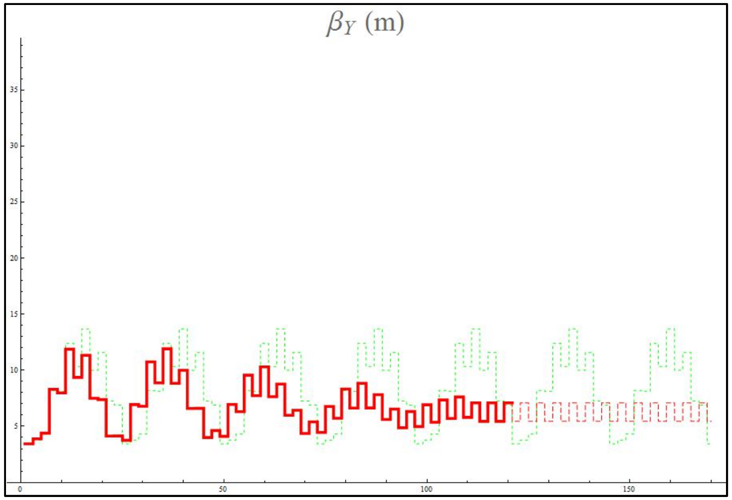
Design $\beta_{X/Y}$

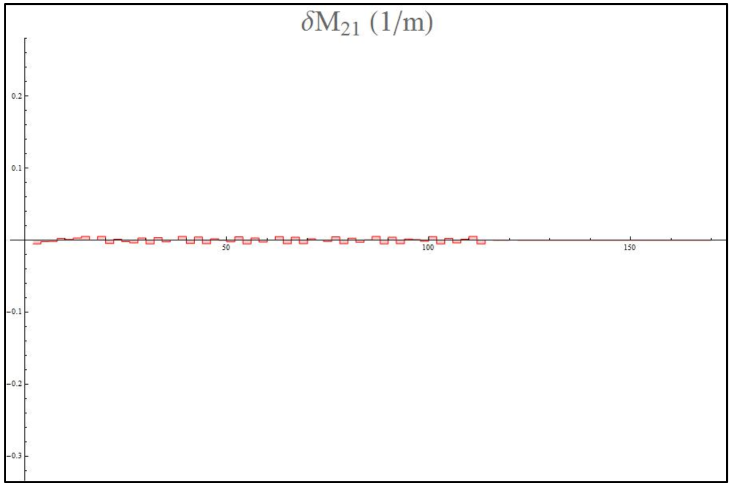
Mismatched

Local Matching

Distributed Matching over 7.5 cells at source







Motivation for Distributed Matching

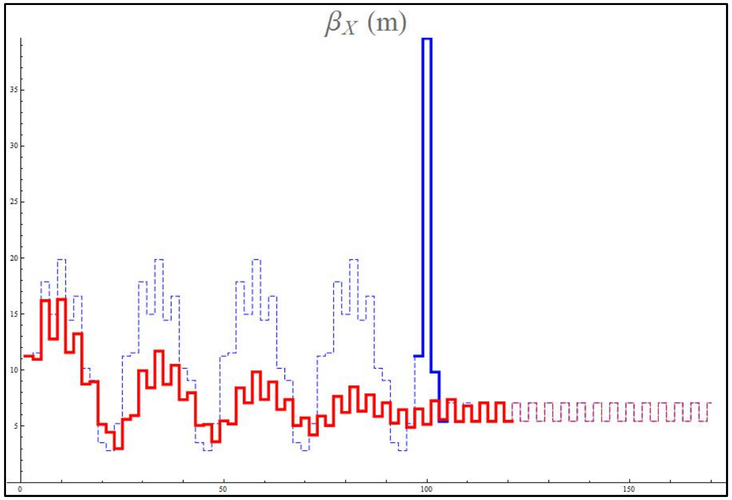
30° FODO Lattice

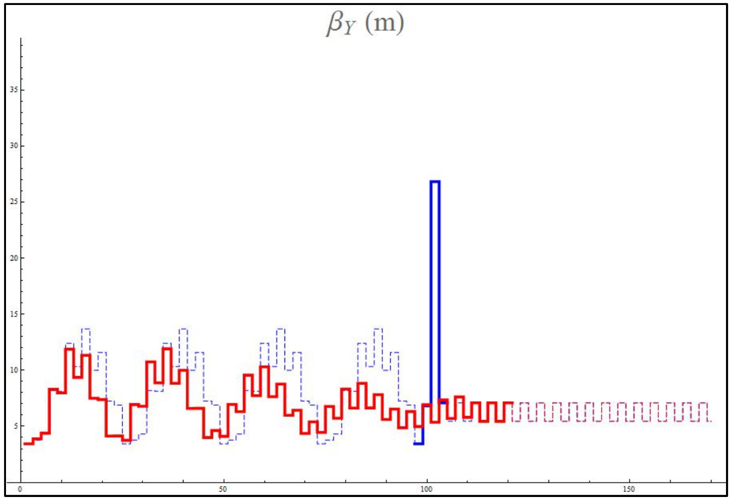
Design $\beta_{X/Y}$

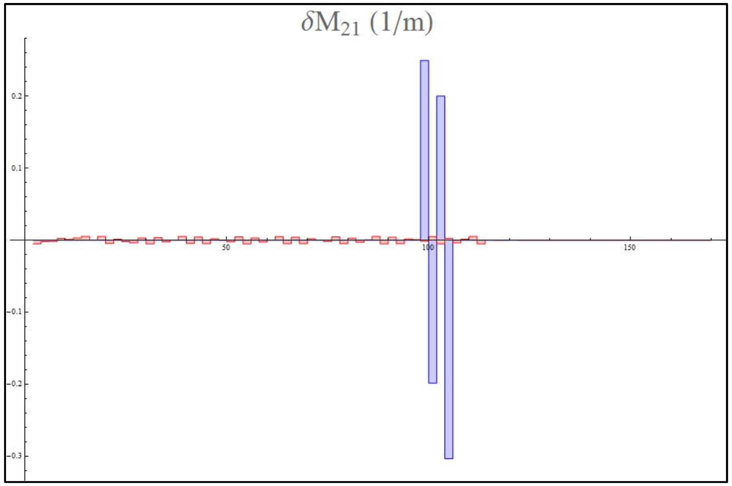
Mismatched

Local Matching

Distributed Matching over 7.5 cells at source







Motivation for Distributed Matching

30° FODO Lattice

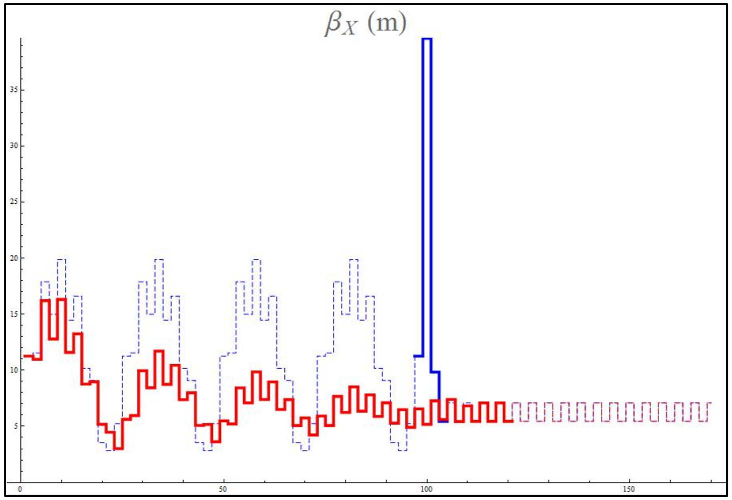
Design $\beta_{X/Y}$

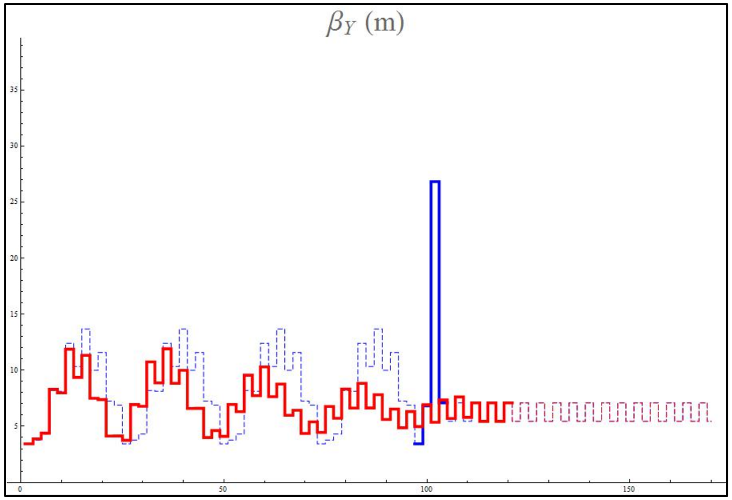
Mismatched

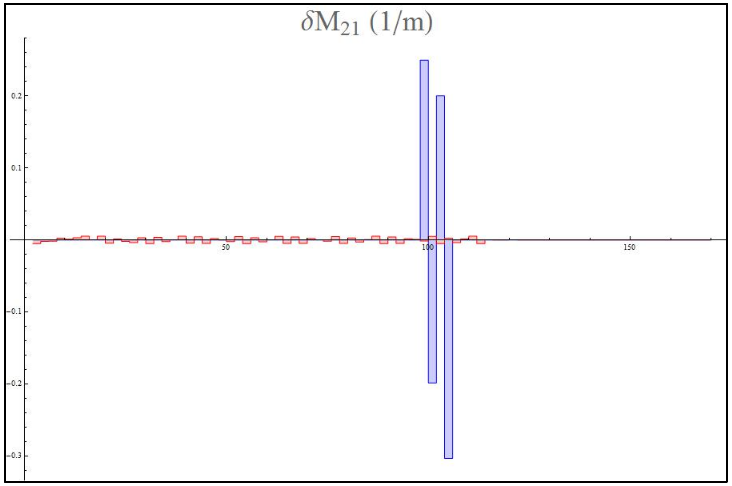
Local Matching

Distributed Matching over 7.5 cells <u>at source</u>

Local Matching with 1% setting Error







Motivation for Distributed Matching

30° FODO Lattice

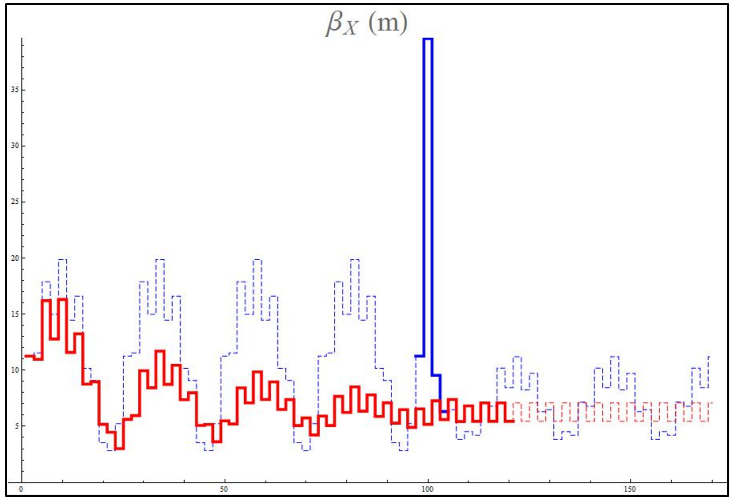
Design $\beta_{X/Y}$

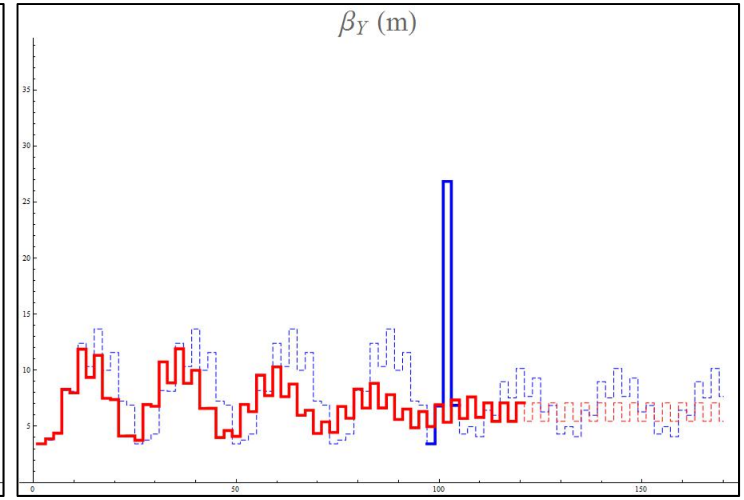
Mismatched

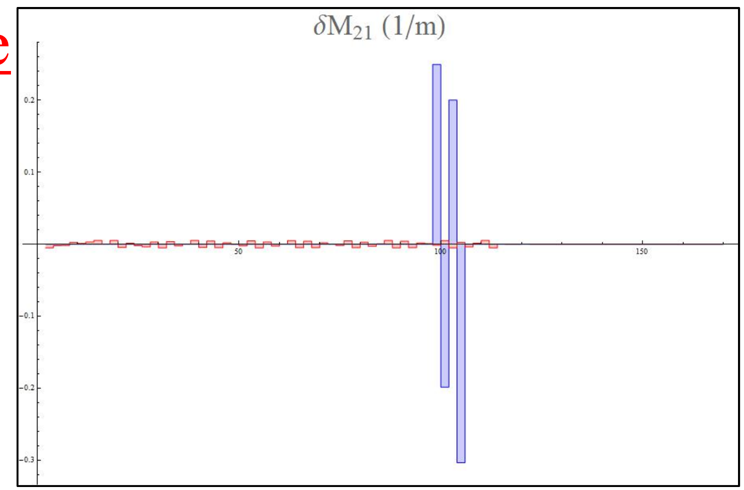
Local Matching

Distributed Matching over 7.5 cells <u>at source</u>

Local Matching with 1% setting Error







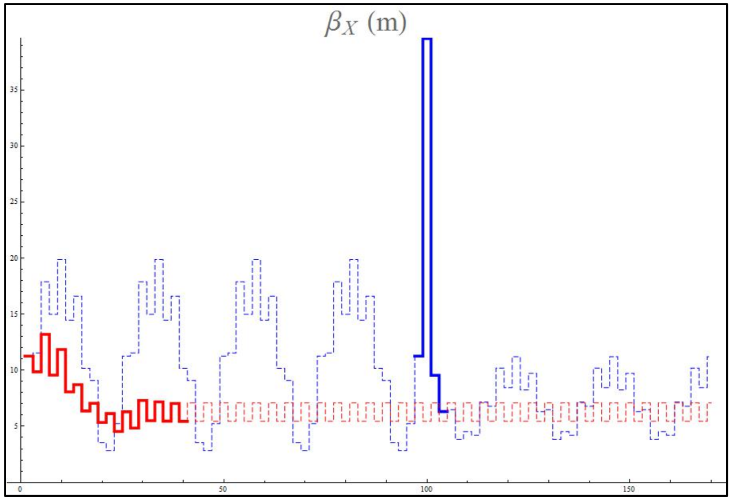
Motivation for Distributed Matching

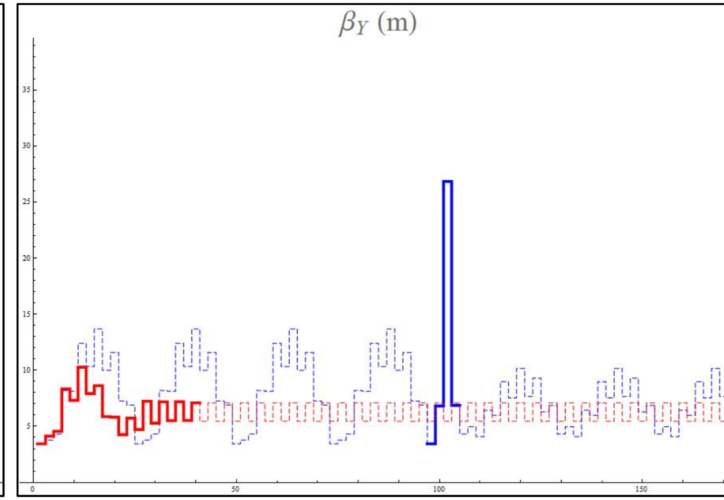
30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching

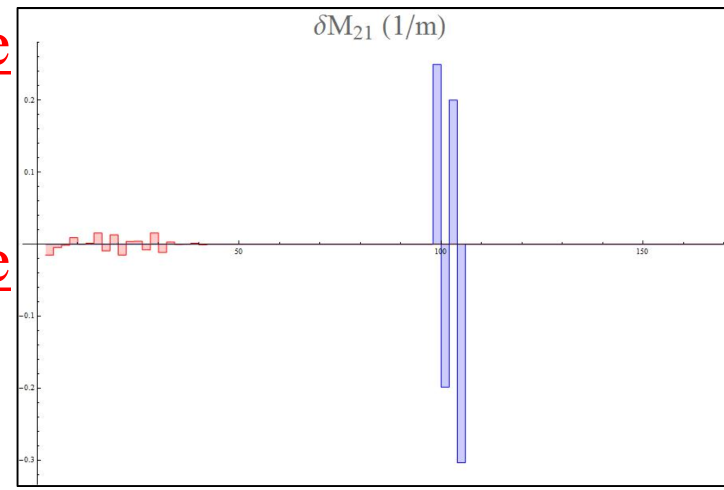




Distributed Matching over 7.5 cells <u>at source</u>

Local Matching with 1% setting Error

Distributed Matching over 2.5 cells <u>at source</u>



Motivation for Distributed Matching

30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

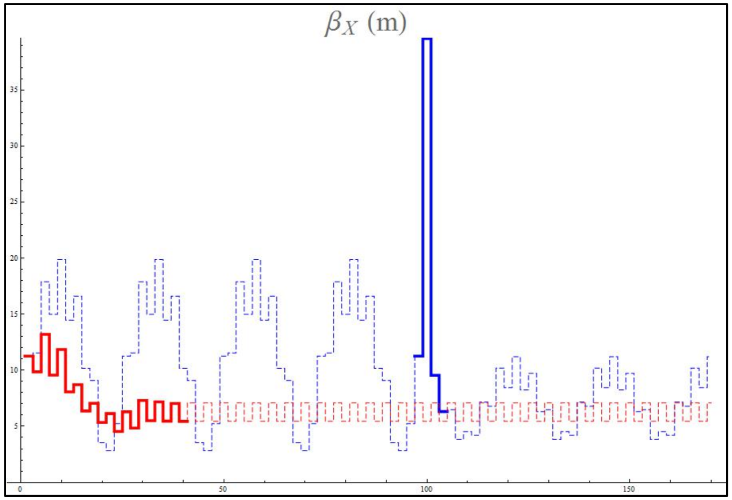
Local Matching

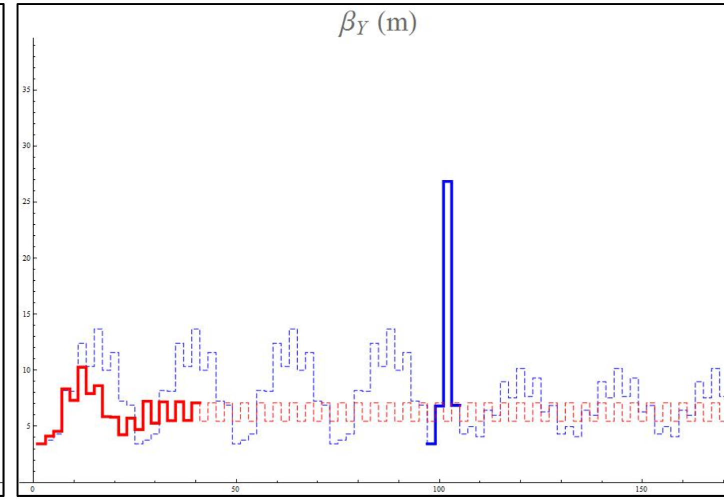
Distributed Matching over 7.5 cells <u>at source</u>

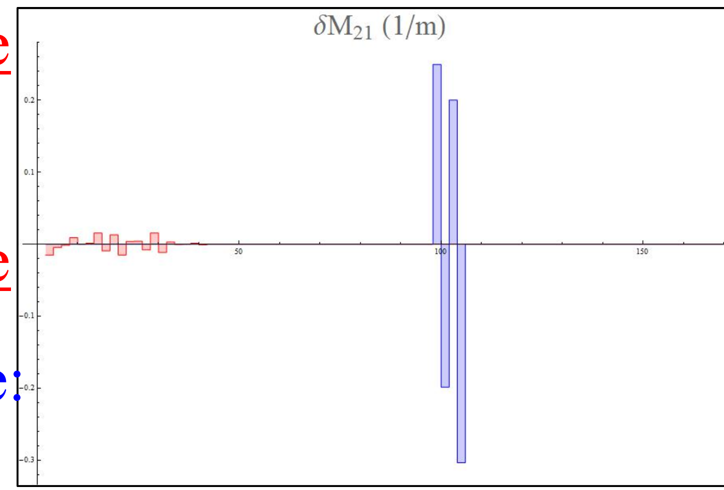
Local Matching with 1% setting Error

Distributed Matching over 2.5 cells <u>at source</u>

Advantages of Distributed Matching Scheme:







Motivation for Distributed Matching

30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching

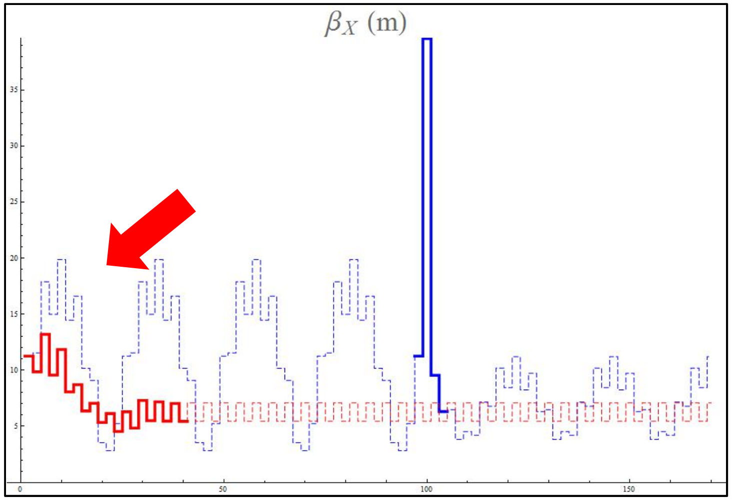
Distributed Matching over 7.5 cells at source

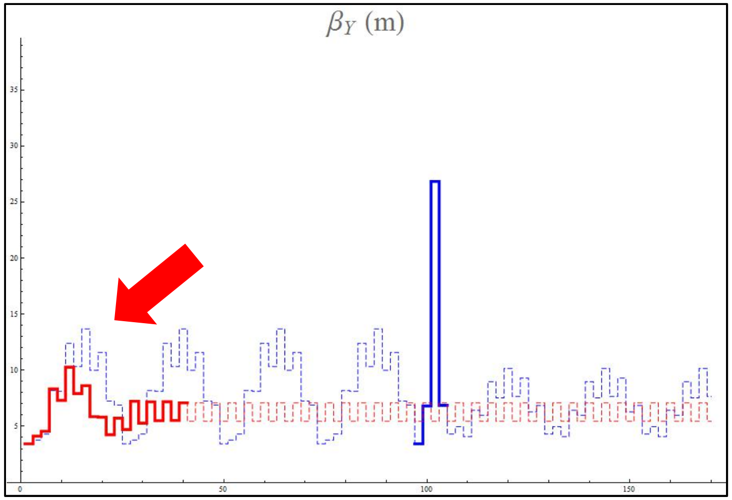
Local Matching with 1% setting Error

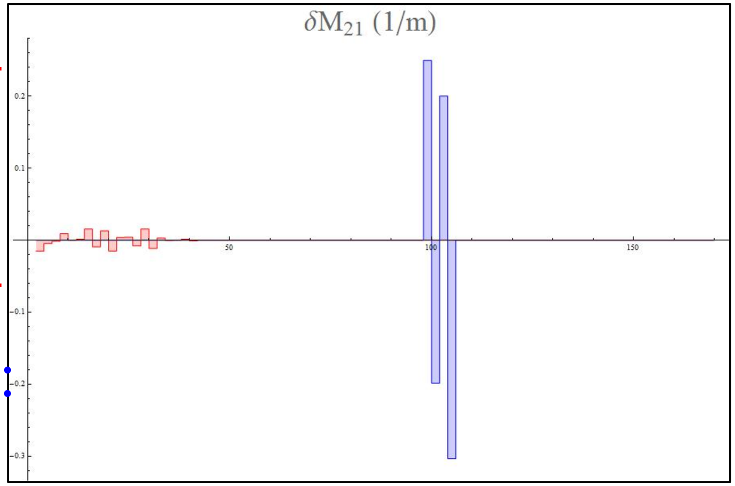
Distributed Matching over 2.5 cells at source

Advantages of Distributed Matching Scheme:

- Mismatch arrested on the sopt







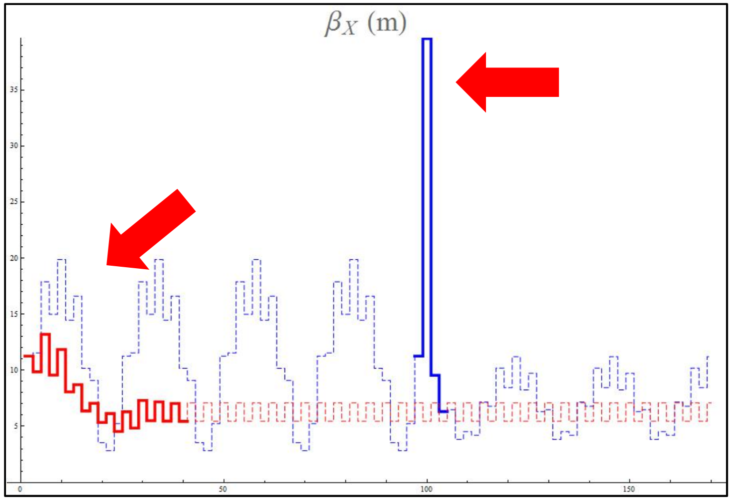
Motivation for Distributed Matching

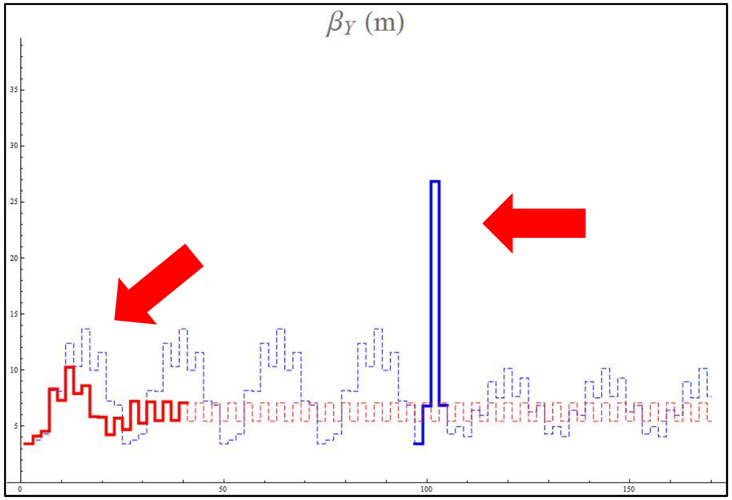
30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching

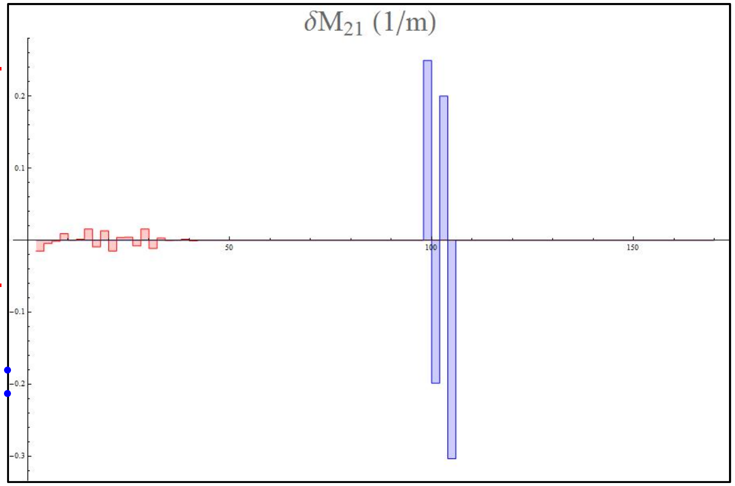




Distributed Matching over 7.5 cells at source

Local Matching with 1% setting Error

Distributed Matching over 2.5 cells at source



Advantages of Distributed Matching Scheme:

- Mismatch arrested on the sopt
- Blowup averted

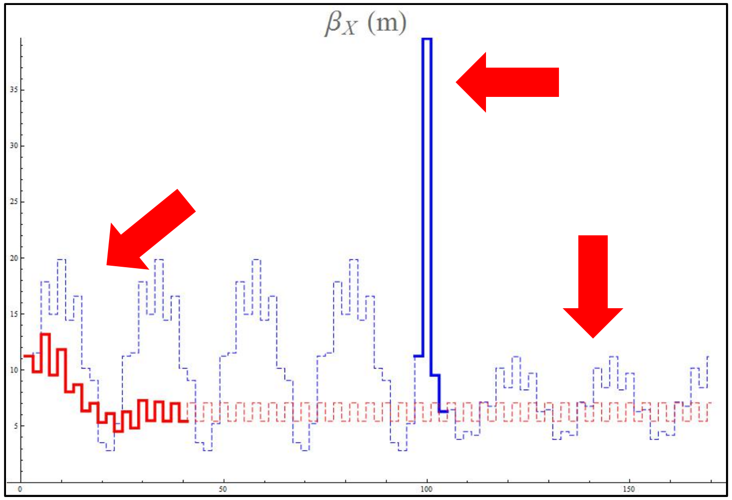
Motivation for Distributed Matching

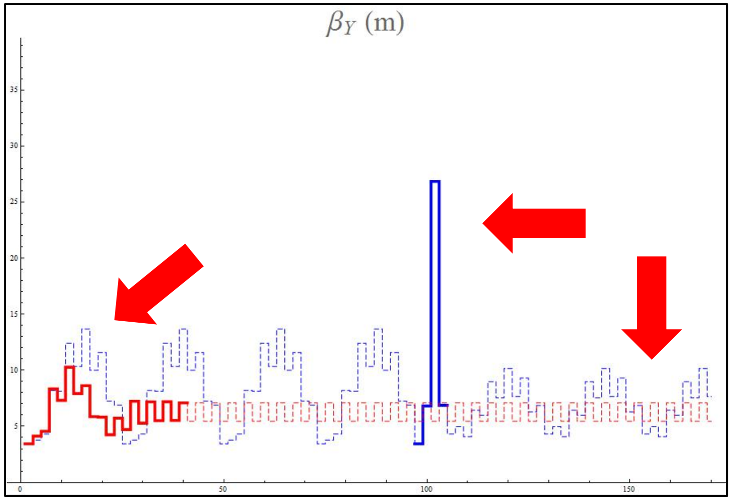
30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching

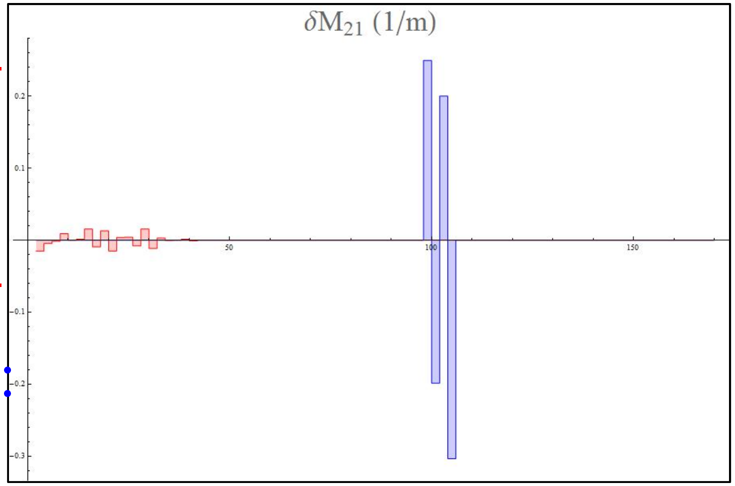




Distributed Matching over 7.5 cells at source

Local Matching with 1% setting Error

Distributed Matching over 2.5 cells at source



Advantages of Distributed Matching Scheme:

- ❖ Mismatch arrested on the sopt
- ❖ Blowup averted
- ❖ Matching failure dynamically corrected

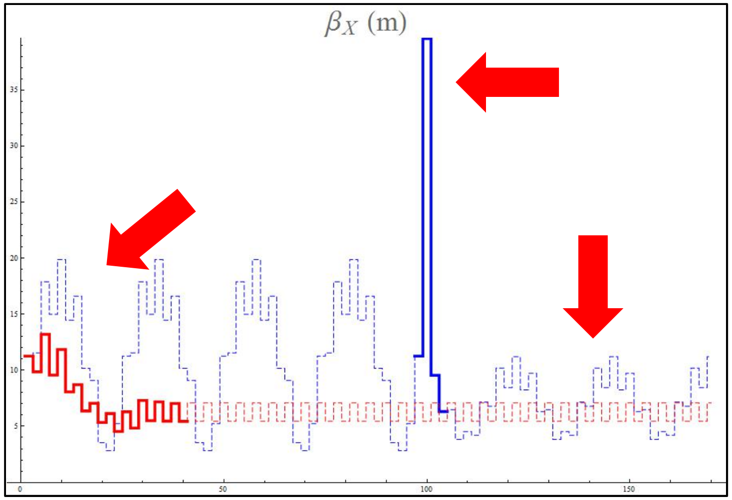
Motivation for Distributed Matching

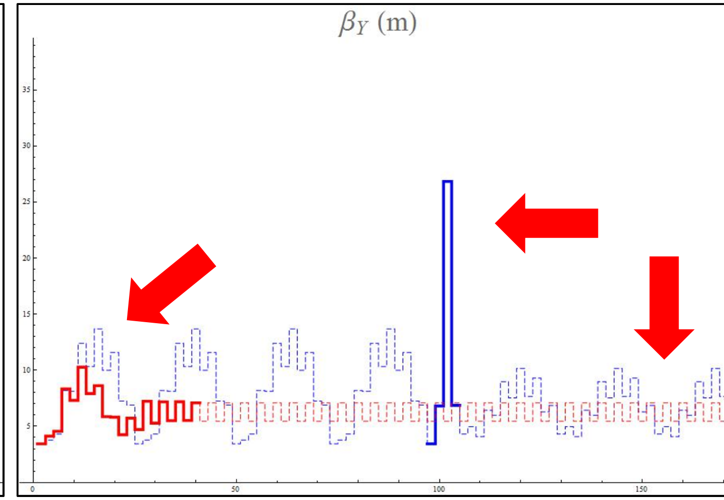
30° FODO Lattice

Design $\beta_{X/Y}$

Mismatched

Local Matching

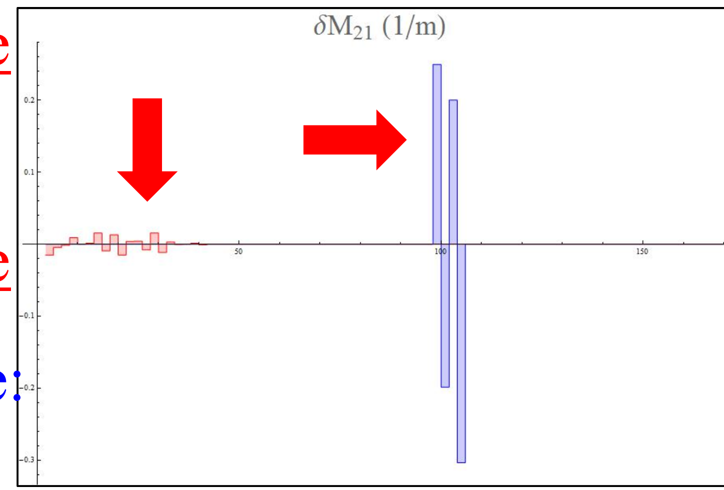




Distributed Matching over 7.5 cells <u>at source</u>

Local Matching with 1% setting Error

Distributed Matching over 2.5 cells <u>at source</u>



Advantages of Distributed Matching Scheme:

- Mismatch arrested on the sopt
- Blowup averted
- Matching failure dynamically corrected
- Baseline optics minimally perturbed

Two Schemes of Accelerator Control Configuration

❖ **Localized Control**	❖ **Distributed Control**
➢ Limited/Costly/Bulky monitor & actuator	➢ Affordable/Compact monitor & actuator
➢ Little cumulative/compounded error	➢ Errors accumulate & compound all over
➢ Damage is mostly localized	➢ Damage happens everywhere
➢ Example: Dispersion, σ_L, Energy, σ_E ….	➢ Example: Transverse orbit

Transverse Matching (Beam &Transport) Fits the Distributed Model Better, But….

Traditionally It Acquired a Local Flavor in Design & Operation.

❖ A legacy deserving revisit: Without adequate monitoring, competent global algorithm, and real time computing power, this was understandable.

❖ Not unlike global steering by correctors before these ingredients were in place.

Aversion to Chaos largely responsible for traditional Local Matching Paradigm

⇒ 100% matching within dedicated section; Other quads passively hold up transport.

➢ Dedicated matching sections are required – Extra constraint on **design** & **operation**

➢ Long range cumulative error; Drastic correction; Local blowup, Solution difficulty.

➢ No recourse to matching failure; No error tolerance; No dynamic correction.

➢ **And:** Large beam envelope/jitter can sample nonlinearities in irreversible ways.

The Message:

➢ Primary role of quads is Envelope/Jitter containment, better actively than passively.

➢ It's not about **whether**, but **how** to use **all** quads for matching in an **intelligent** way.

Optimization ⇒ Optimized Trade-Off – (2nd) Alternate View

- Distributed Matching ⇒ Partially Matched Solutions
  ⇒ Degeneracy ⇒ Require Further Constraints to Produce Unique Answer
- Matching is never single-objective at the cost to all else (e.g. Quad Strength)
  ⇒ Needs Rigorous Framework for Trade-Off between Competing Objectives

Lagrange Multiplier Formulation

Variables: $k_1, k_2, \ldots k_N$

Objective 1: $\boldsymbol{F}(k_1, k_2, \ldots k_N)$

Objective 2: $\boldsymbol{H}(k_1, k_2, \ldots k_N)$

Optimal Local Trade-Off

⇒

$$\begin{cases} \nabla F = \lambda \cdot \nabla H \\ H = h \end{cases} \Rightarrow F = f(h)$$

Optimization ⇒ Optimized Trade-Off – (2nd) Alternate View

- ❖ Distributed Matching ⇒ Partially Matched Solutions
  ⇒ Degeneracy ⇒ Require Further Constraints to Produce Unique Answer
- ❖ Matching is never single-objective at the cost to all else (e.g. Quad Strength)
  ⇒ Needs Rigorous Framework for Trade-Off between Competing Objectives

Lagrange Multiplier Formulation

Variables: $k_1, k_2, \dots k_N$

Objective 1: $\boldsymbol{F}(k_1, k_2, \dots k_N)$

Objective 2: $\boldsymbol{H}(k_1, k_2, \dots k_N)$

Optimal Local Trade-Off ➡

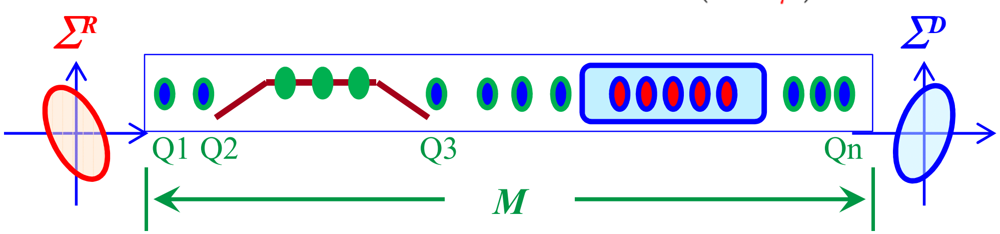
$$\begin{cases} \nabla F = \lambda \cdot \nabla H \\ F = f \end{cases} \Rightarrow H = h(f)$$

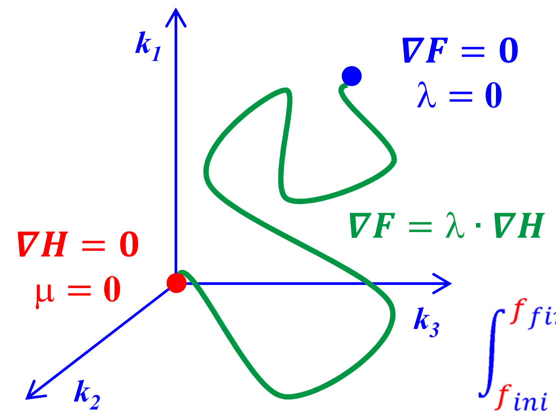
Tangency Condition

4D Matching Problem

$$\Sigma = \begin{pmatrix} \langle X\,X \rangle & \langle X\,X' \rangle \\ \langle X\,X' \rangle & \langle X'X' \rangle \end{pmatrix} = \begin{pmatrix} \beta & -\alpha \\ -\alpha & \gamma \end{pmatrix} \cdot \epsilon$$

R: Actual
D: Design





$$\int_{f_{ini}}^{f_{fin}} df \, \frac{d\boldsymbol{k}}{df} \; ?$$

Objective 1: $F = \Phi$
Generalized 4D mismatch factor

$$\boldsymbol{F} = \Phi = \frac{1}{4} Tr\left(\Sigma_D^{-1} \cdot M(k_m) \cdot \Sigma_R \cdot M^T(k_m)\right) \geq 1$$

R: Actual
D: Design

Objective 2: $H = \Delta K$:
RMS Quad deviation off design

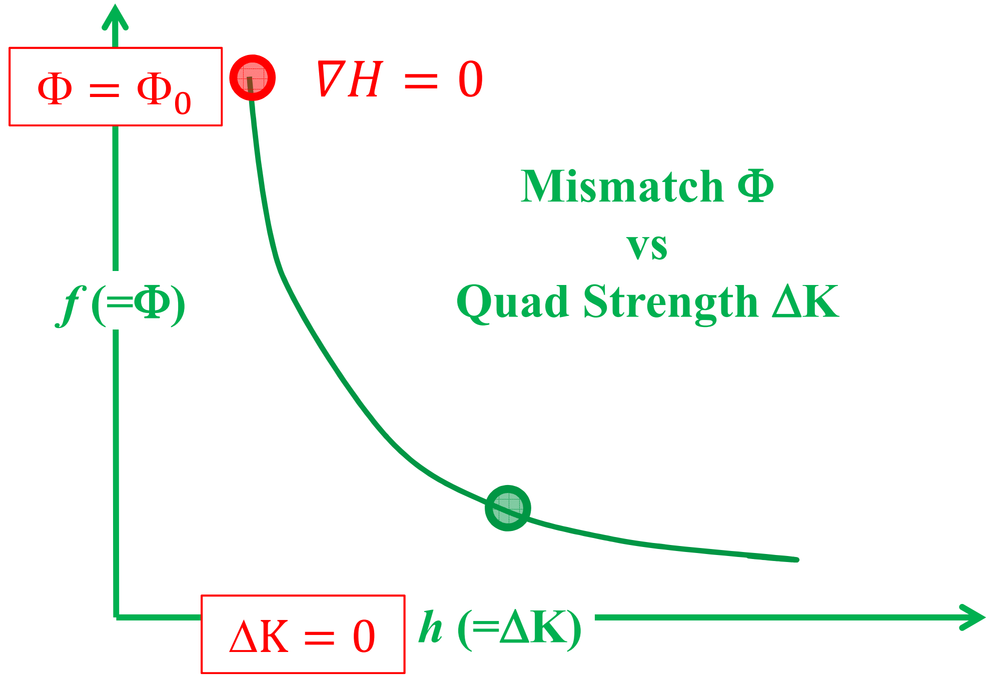
$$\boldsymbol{H} = \Delta K = \sum_{m=1}^{N_Q} (k_m^R - k_m^D)^2 = \sum_{k=1}^{N_Q} \delta k_m{}^2 \geq 0$$

RECIPE

❖ **Starting point ($\Delta K=0, \Phi = \Phi_0$):**

$\mu=0; \quad k_i=0$

$$\left.\frac{d\boldsymbol{k_i}}{d\mu}\right|_{\mu=0} = \frac{1}{2}\left.\frac{\partial F(\boldsymbol{k})}{\partial k_i}\right|_{k_m=0}$$



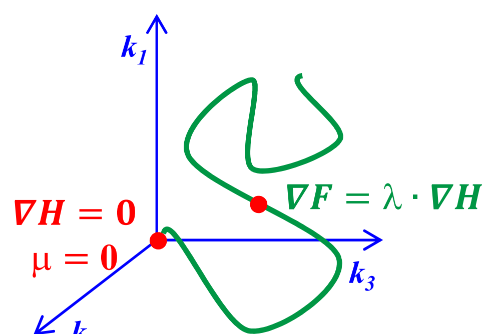
❖ **Evolution of k_i ($\lambda = 1/\mu$):**

$$\left.\frac{d\boldsymbol{k}}{d\boldsymbol{f}}\right| = \frac{1}{\lambda}\cdot\frac{Adj(\boldsymbol{M})\cdot\boldsymbol{R}}{\boldsymbol{R}^T\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R}}, \quad \left.\frac{d\boldsymbol{k}}{d\boldsymbol{h}}\right| = \frac{Adj(\boldsymbol{M})\cdot\boldsymbol{R}}{\boldsymbol{R}^T\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R}} \qquad \boldsymbol{R}^T\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R}\neq 0$$

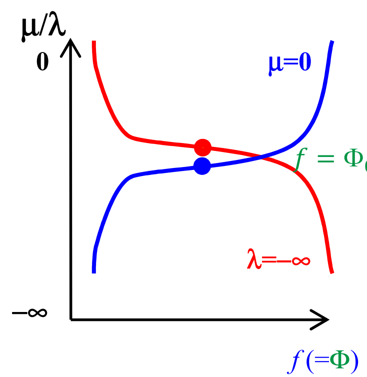
$$\left.\frac{d\boldsymbol{k}}{d\lambda}\right| = \boldsymbol{M}^{-1}\cdot\boldsymbol{R}, \quad \left.\frac{d\boldsymbol{k}}{d\mu}\right| = \boldsymbol{N}^{-1}\cdot\boldsymbol{S} \qquad Det(\boldsymbol{M})\neq 0$$

$$\boldsymbol{M}_{ij} = \frac{\partial^2(F(\boldsymbol{k}) - \lambda\cdot H(\boldsymbol{k}))}{\partial k_i\partial k_j}, \qquad \boldsymbol{R}_i = \frac{\partial H(\boldsymbol{k})}{\partial k_i}$$

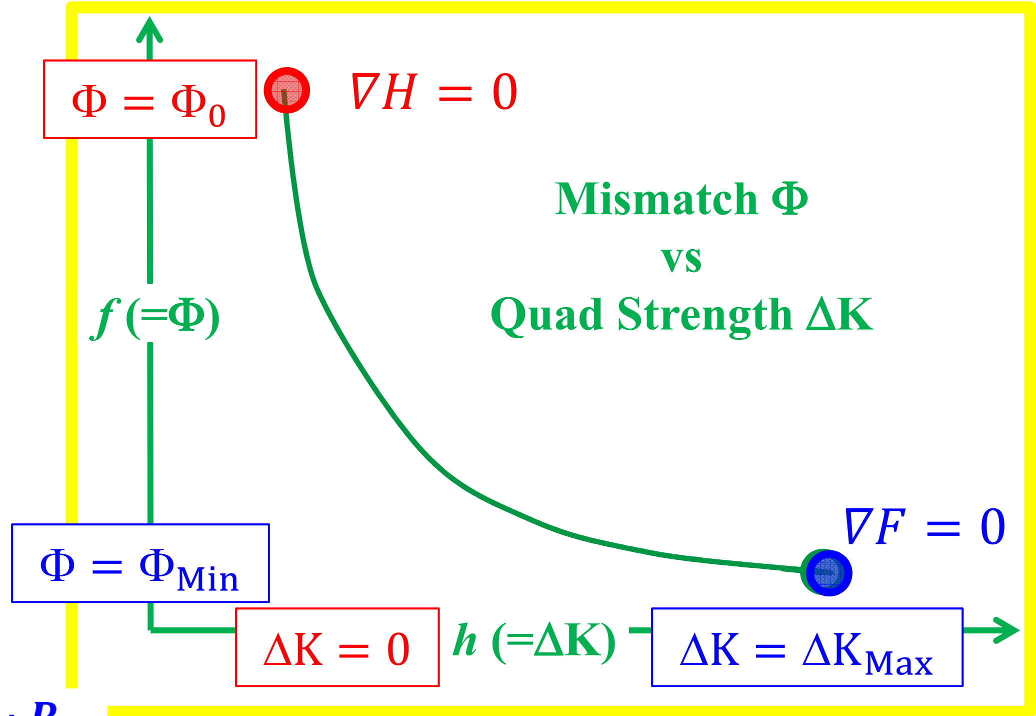
Evolution of Quad k_i



Evolution of λ & $\mu = 1/\lambda$ vs Φ



RECIPE



- **Starting point ($\Delta K=0, \Phi = \Phi_0$):**

  $\mu=0; \quad k_i=0$

$$\left.\frac{d\boldsymbol{k_i}}{d\mu}\right|_{\mu=0} = \frac{1}{2}\left.\frac{\partial F(\boldsymbol{k})}{\partial k_i}\right|_{k_m=0}$$

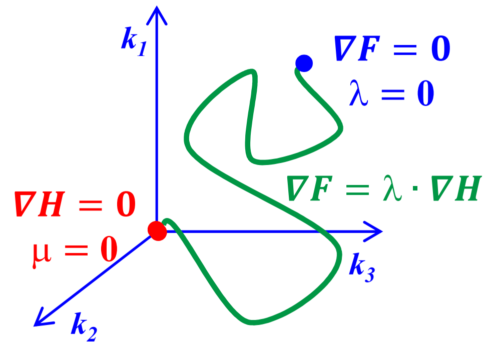
- **Evolution of k_i ($\lambda = 1/\mu$):**

$$\left.\frac{d\boldsymbol{k}}{d\boldsymbol{f}}\right| = \frac{1}{\lambda}\cdot\frac{Adj(\boldsymbol{M})\cdot\boldsymbol{R}}{\boldsymbol{R}^T\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R}}, \quad \left.\frac{d\boldsymbol{k}}{d\boldsymbol{h}}\right| = \frac{Adj(\boldsymbol{M})\cdot\boldsymbol{R}}{\boldsymbol{R}^T\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R}} \qquad \boldsymbol{R}^T\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R} \neq 0$$

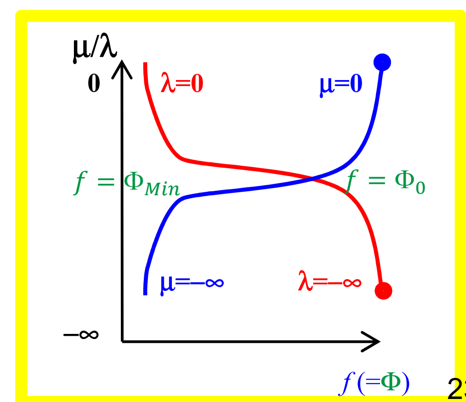
$$\left.\frac{d\boldsymbol{k}}{d\lambda}\right| = \boldsymbol{M}^{-1}\cdot\boldsymbol{R}, \quad \left.\frac{d\boldsymbol{k}}{d\mu}\right| = \boldsymbol{N}^{-1}\cdot\boldsymbol{S} \qquad Det(\boldsymbol{M}) \neq 0$$

$$\boldsymbol{M}_{ij} = \frac{\partial^2(F(\boldsymbol{k}) - \lambda\cdot H(\boldsymbol{k}))}{\partial k_i \partial k_j}, \qquad \boldsymbol{R}_i = \frac{\partial H(\boldsymbol{k})}{\partial k_i}$$

Evolution of Quad k_i



Evolution of λ & $\mu=1/\lambda$ vs Φ



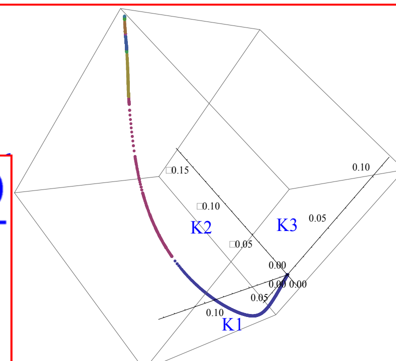
- **End Point (Optimally Matched):**

  $\lambda=0$

More Robust Formulation (Singularity Free)

❖ Tailored to a Runge-Kutta type process with only local derivatives defined.

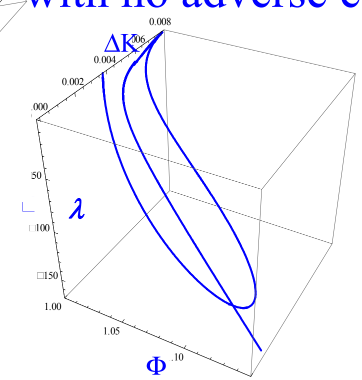
Start/Stop	Integration Formula	Evolution of Competing Objectives
$0 > \mu > -1$	$\frac{d\boldsymbol{k}}{dk} = \pm\widehat{\boldsymbol{Q}}, \quad \boldsymbol{Q} = Adj(\boldsymbol{N}) \cdot \boldsymbol{S}$ $\frac{d\mu}{dk} = \pm\frac{Det(\boldsymbol{N})}{\lvert\boldsymbol{Q}\rvert}$	$\frac{df}{dk} = \pm\frac{(\boldsymbol{S}^T \cdot Adj(\boldsymbol{N}) \cdot \boldsymbol{S})}{\lvert\boldsymbol{Q}\rvert} = \pm\boldsymbol{S}^T \cdot \widehat{\boldsymbol{Q}}$ $\frac{dh}{dk} = \pm\frac{\mu \cdot (\boldsymbol{S}^T \cdot Adj(\boldsymbol{N}) \cdot \boldsymbol{S})}{\lvert\boldsymbol{Q}\rvert} = \pm\mu \cdot \boldsymbol{S}^T \cdot \widehat{\boldsymbol{Q}}$



with no adverse effects.

$$\boldsymbol{M}_{ij} = \frac{\partial^2(F(\boldsymbol{k}) - \lambda \cdot H(\boldsymbol{k}))}{\partial k_i \partial k_j}, \qquad \boldsymbol{R}_i = \frac{\partial H(\boldsymbol{k})}{\partial k_i}$$

$$\boldsymbol{N}_{ij} = \frac{\partial^2(H(\boldsymbol{k}) - \mu \cdot F(\boldsymbol{k}))}{\partial k_i \partial k_j}, \qquad \boldsymbol{S}_i = \frac{\partial F(\boldsymbol{k})}{\partial k_i}$$



ion

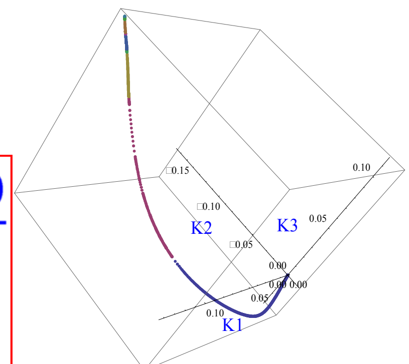
More Robust Formulation (Singularity Free)

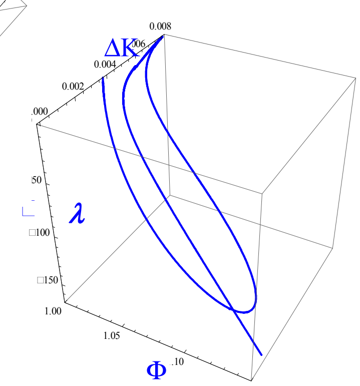
❖ Tailored to a Runge-Kutta type process with only local derivatives defined.

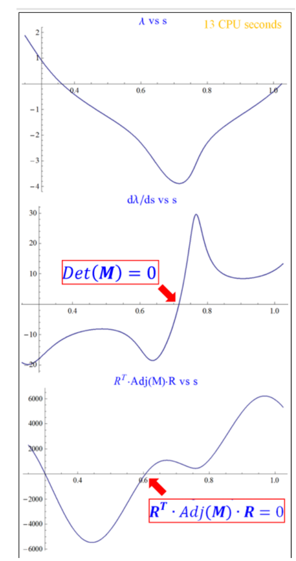
Start/Stop	Integration Formula	Evolution of Competing Objectives
$0 > \mu > -1$	$\frac{d\boldsymbol{k}}{dk} = \pm\widehat{\boldsymbol{Q}}, \quad \boldsymbol{Q} = Adj(\boldsymbol{N})\cdot\boldsymbol{S}$ $\frac{d\mu}{dk} = \pm\frac{Det(\boldsymbol{N})}{\lvert\boldsymbol{Q}\rvert}$	$\frac{df}{dk} = \pm\frac{(\boldsymbol{S}^{\boldsymbol{T}}\cdot Adj(\boldsymbol{N})\cdot\boldsymbol{S})}{\lvert\boldsymbol{Q}\rvert} = \pm\boldsymbol{S}^{\boldsymbol{T}}\cdot\widehat{\boldsymbol{Q}}$ $\frac{dh}{dk} = \pm\frac{\mu\cdot(\boldsymbol{S}^{\boldsymbol{T}}\cdot Adj(\boldsymbol{N})\cdot\boldsymbol{S})}{\lvert\boldsymbol{Q}\rvert} = \pm\mu\cdot\boldsymbol{S}^{\boldsymbol{T}}\cdot\widehat{\boldsymbol{Q}}$
$-1 < \lambda < 0$	$\frac{d\boldsymbol{k}}{dk} = \pm\widehat{\boldsymbol{P}}, \quad \boldsymbol{P} = Adj(\boldsymbol{M})\cdot\boldsymbol{R}$ $\frac{d\lambda}{dk} = \pm\frac{Det(\boldsymbol{M})}{\lvert\boldsymbol{P}\rvert}$	$\frac{df}{dk} = \pm\frac{\lambda\cdot(\boldsymbol{R}^{\boldsymbol{T}}\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R})}{\lvert\boldsymbol{P}\rvert} = \pm\lambda\cdot\boldsymbol{R}^{\boldsymbol{T}}\cdot\widehat{\boldsymbol{P}}$ $\frac{dh}{dk} = \pm\frac{(\boldsymbol{R}^{\boldsymbol{T}}\cdot Adj(\boldsymbol{M})\cdot\boldsymbol{R})}{\lvert\boldsymbol{P}\rvert} = \pm\boldsymbol{R}^{\boldsymbol{T}}\cdot\widehat{\boldsymbol{P}}$

$$\boldsymbol{M}_{\boldsymbol{ij}} = \frac{\partial^2(F(\boldsymbol{k}) - \lambda\cdot H(\boldsymbol{k}))}{\partial k_i \partial k_j}, \qquad \boldsymbol{R}_{\boldsymbol{i}} = \frac{\partial H(\boldsymbol{k})}{\partial k_i}$$

$$\boldsymbol{N}_{\boldsymbol{ij}} = \frac{\partial^2(H(\boldsymbol{k}) - \mu\cdot F(\boldsymbol{k}))}{\partial k_i \partial k_j}, \qquad \boldsymbol{S}_{\boldsymbol{i}} = \frac{\partial F(\boldsymbol{k})}{\partial k_i}$$







Determinism – What Makes This Algorithm Unique

- Deterministic Start-of-Procedure
  - User defined starting k_i (e.g. $\Delta k_i = 0$, and $dk_i/d\mu$ accordingly)
  - No "inspired guesses" for initial value
  - No random number search
  - No case-by-case parameter tweaking to "guide" the solution

- Deterministic End-of-Procedure

  A. **If $\lambda = 0$, Stop.** (Best matching when $\Phi=1$ is not rigorously possible)

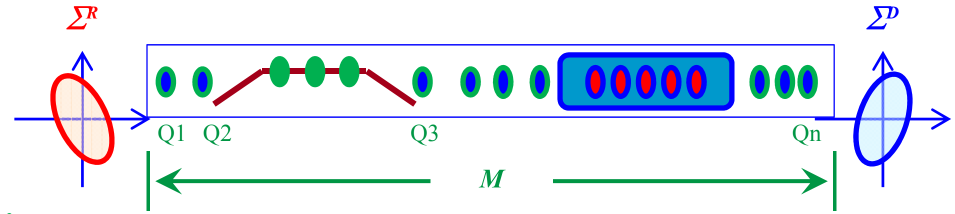
  B. **If $\lambda \neq 0$, Don't stop.** (Big gain by insisting on $\lambda = 0$ even when $\Phi \cong 1$)

  - Both are less trivial than appear
  - Conventional algorithm: Ambivalent about **A**, and can stop short of **B** and miss significant payoff. (Example to follow)

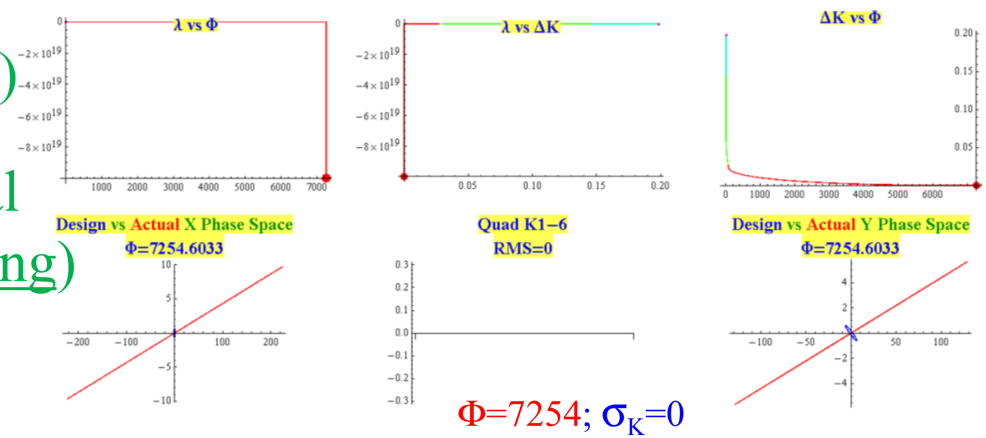
- A Solution is Guaranteed, Plus
  - Guaranteed Global optimum for all intermediate solutions
  - Entire range of intermediate optimal solutions between $\mu=0$ and $\lambda=0$

More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a slow function of optics/system complexity
- Systematic procedure to map out and isolate Global optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for distributed matching)

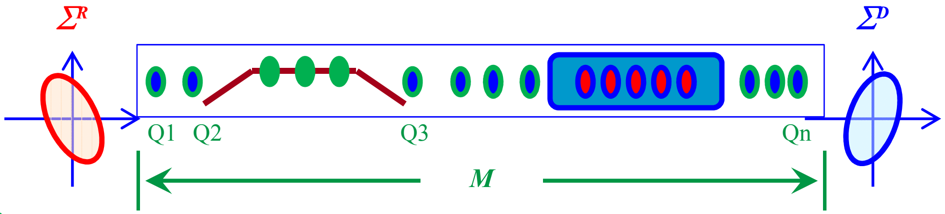


Try Solving $\nabla\Phi = 0$ for 5 Quads

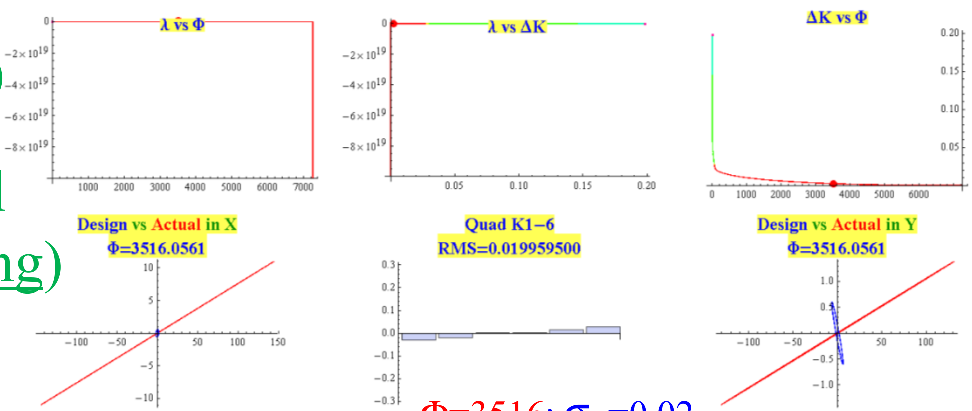


More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a <u>slow</u> function of optics/system complexity
- Systematic procedure to map out and isolate <u>Global</u> optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for <u>distributed matching</u>)



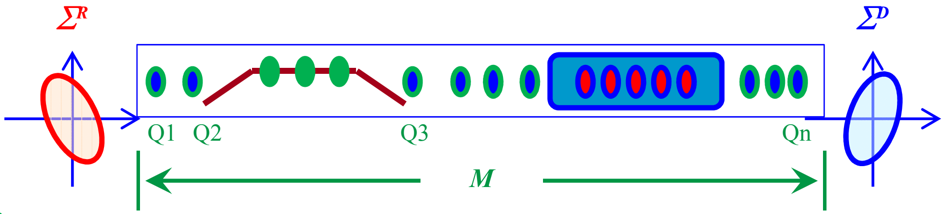
Try Solving ∇Φ =0 for 5 Quads



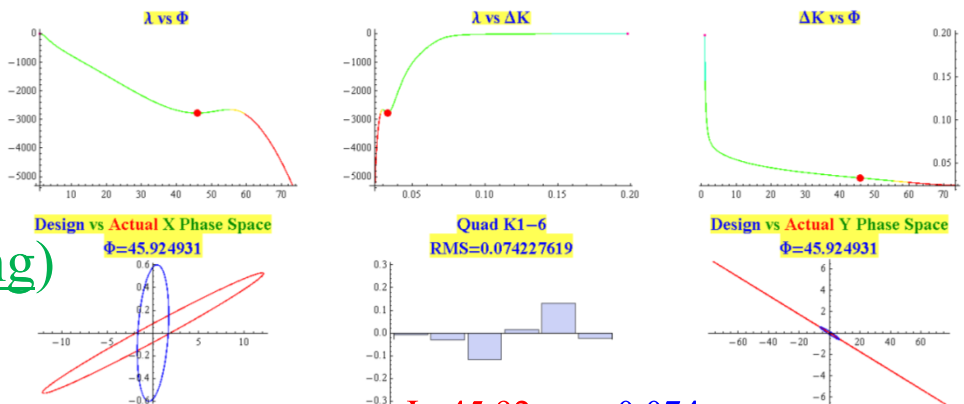
$\Phi=3516$; $\sigma_K=0.02$

More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a <u>slow</u> function of optics/system complexity
- Systematic procedure to map out and isolate <u>Global</u> optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for <u>distributed matching</u>)



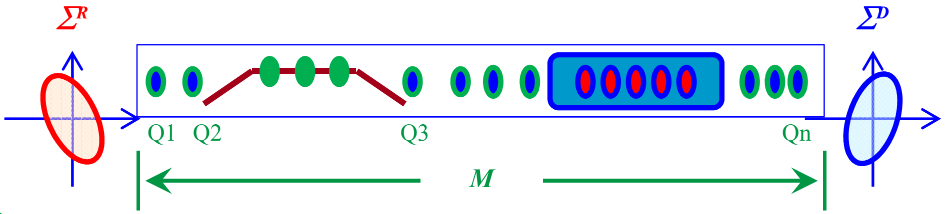
Try Solving $\nabla\Phi = 0$ for 5 Quads



$\Phi=45.92$; $\sigma_K=0.074$

More Advantages

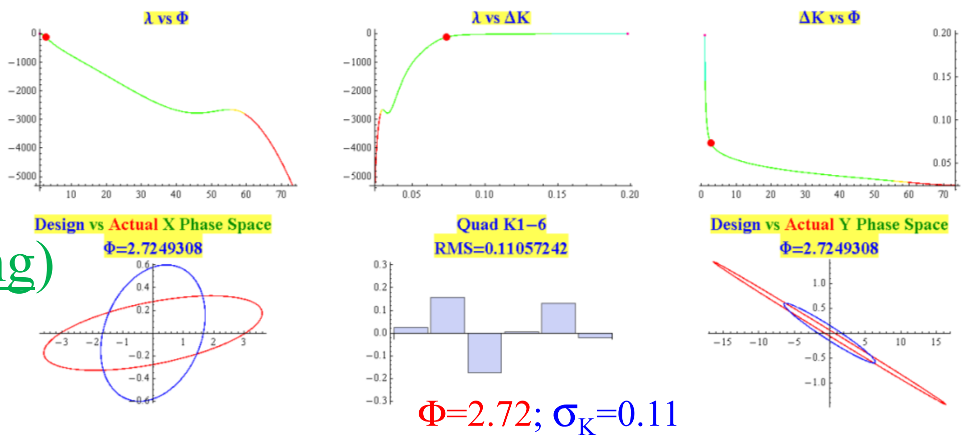
- Works on any system, including XY-coupled and interspersed modules



- Computational demand is a slow function of optics/system complexity

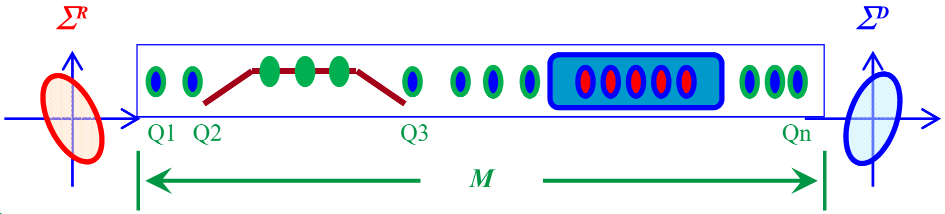
Try Solving $\nabla\Phi = 0$ for 5 Quads

- Systematic procedure to map out and isolate Global optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for distributed matching)

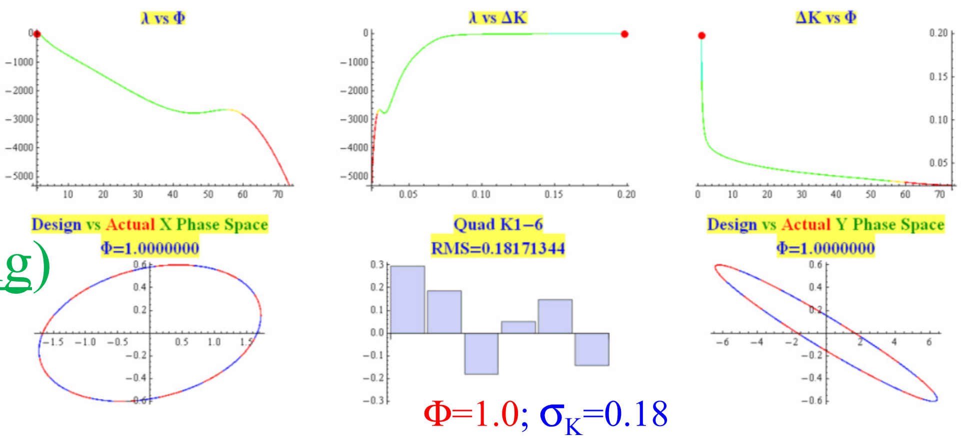


More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a slow function of optics/system complexity
- Systematic procedure to map out and isolate Global optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for distributed matching)

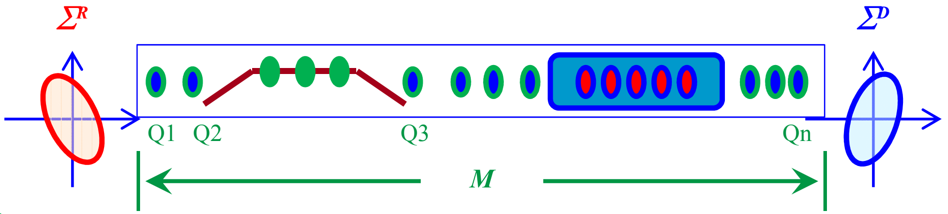


Try Solving $\nabla\Phi = 0$ for 5 Quads

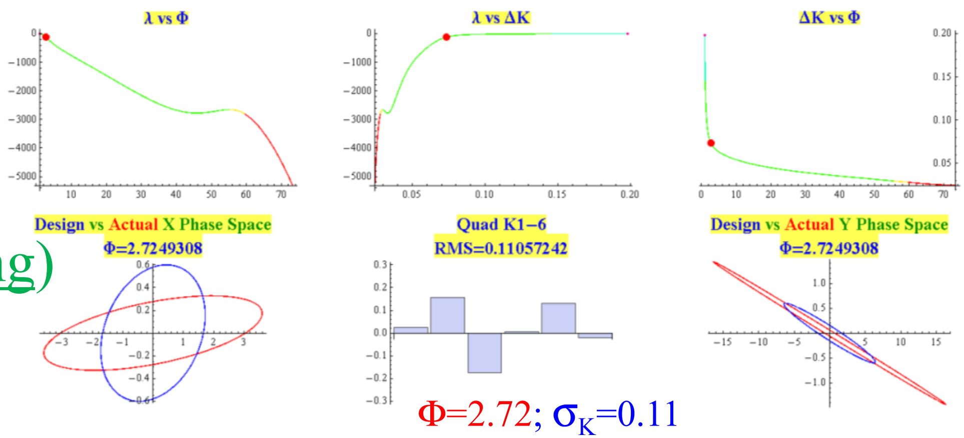


More Advantages

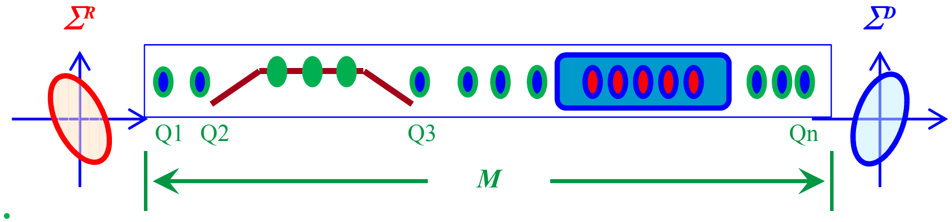
- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a slow function of optics/system complexity
- Systematic procedure to map out and isolate Global optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for distributed matching)



Try Solving $\nabla\Phi = 0$ for 5 Quads

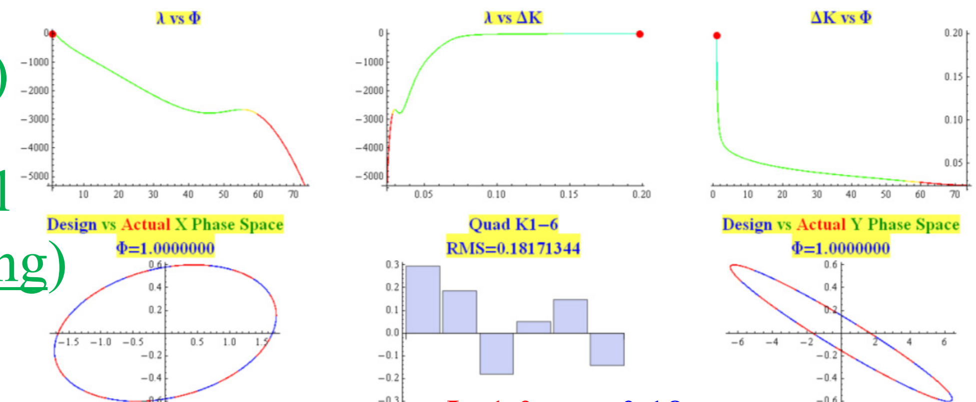


More Advantages



- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a <u>slow</u> function of optics/system complexity
- Systematic procedure to map out and isolate <u>Global</u> optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for <u>distributed matching</u>)

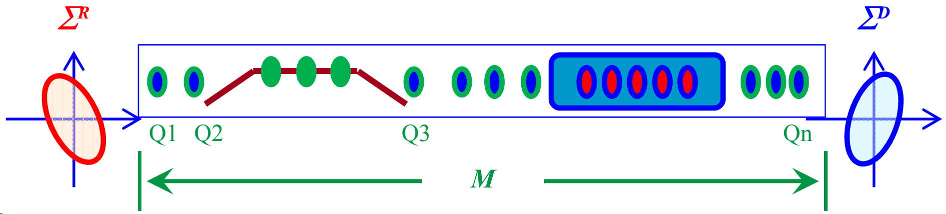
Try Solving $\nabla\Phi = 0$ for 5 Quads



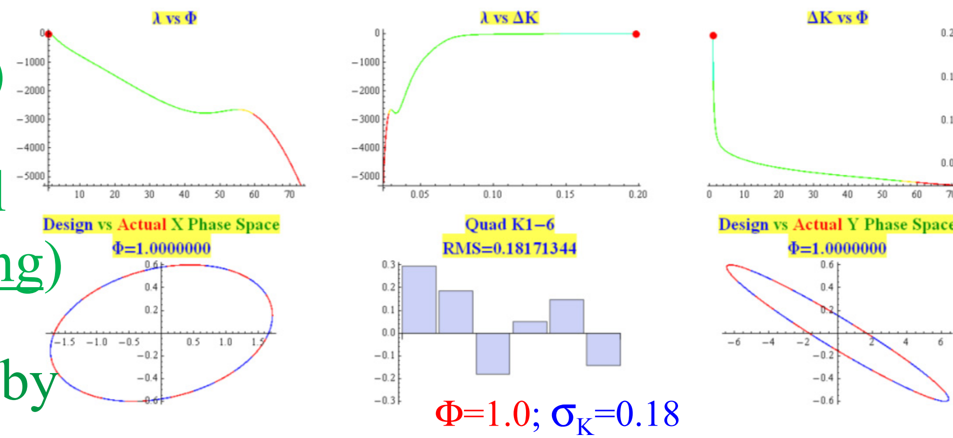
$\Phi=1.0$; $\sigma_K=0.18$

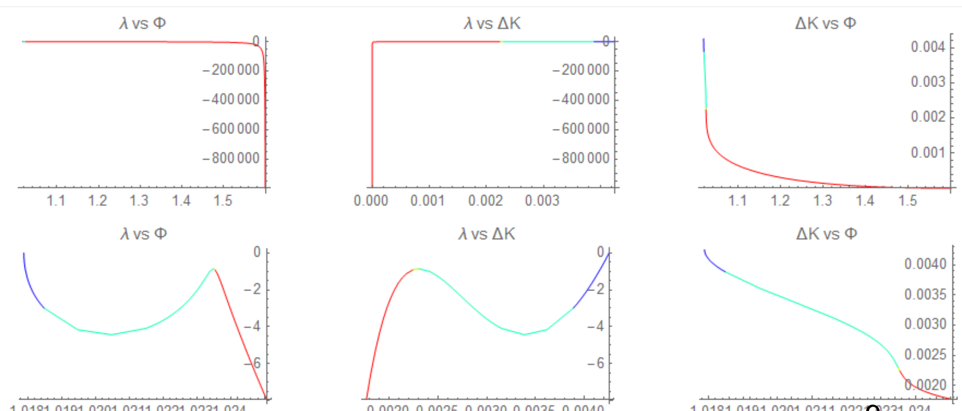
More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a <u>slow</u> function of optics/system complexity
- Systematic procedure to map out and isolate <u>Global</u> optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for <u>distributed matching</u>)
- Points of diminished return identified by well-defined procedure (Det(M)=0)



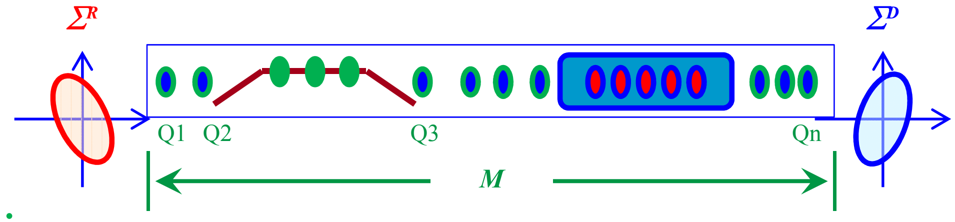
Try Solving $\nabla\Phi = 0$ for 5 Quads



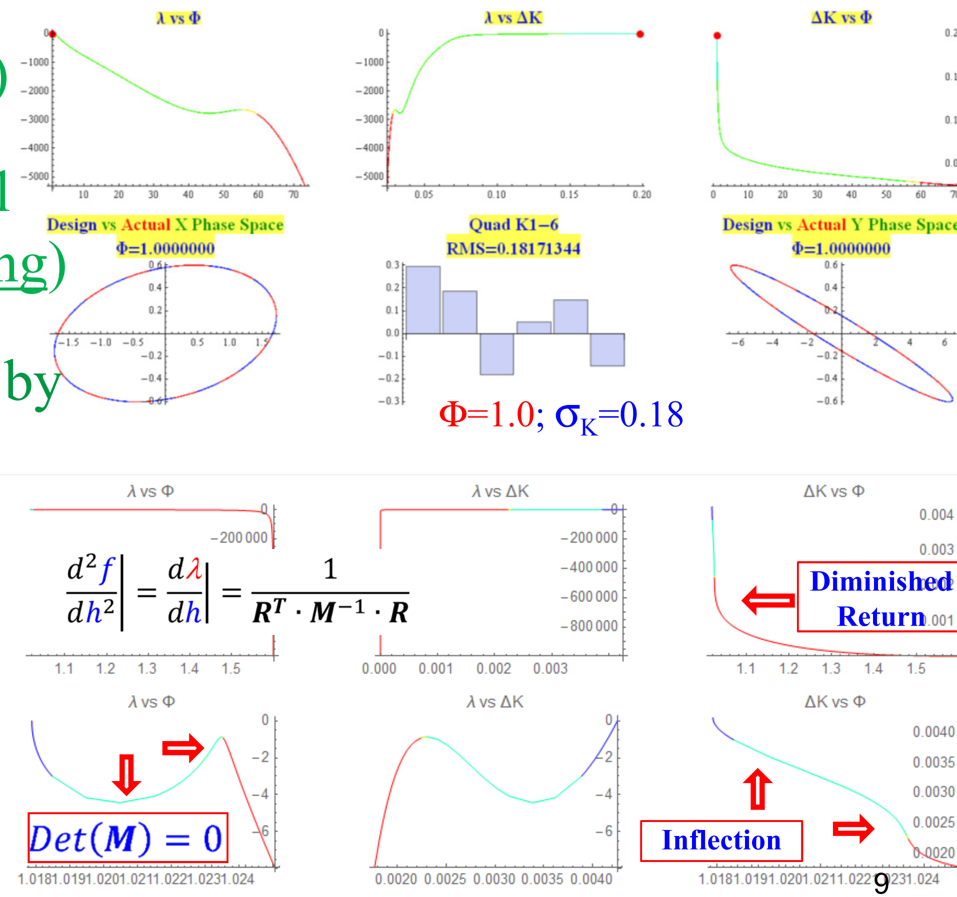


More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a slow function of optics/system complexity
- Systematic procedure to map out and isolate Global optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for distributed matching)
- Points of diminished return identified by well-defined procedure (Det(M)=0)

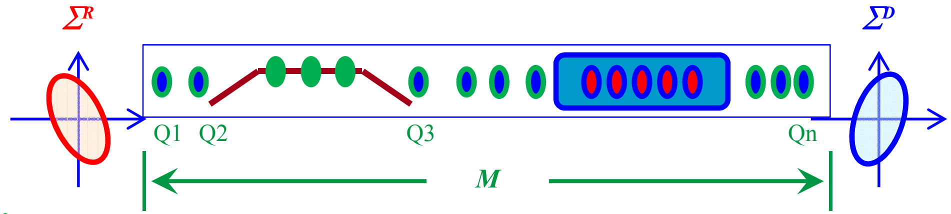


Try Solving ∇Φ =0 for 5 Quads

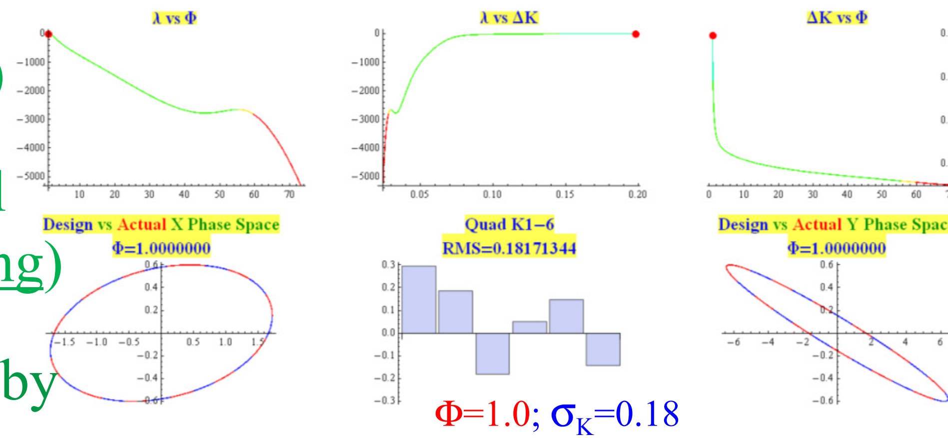


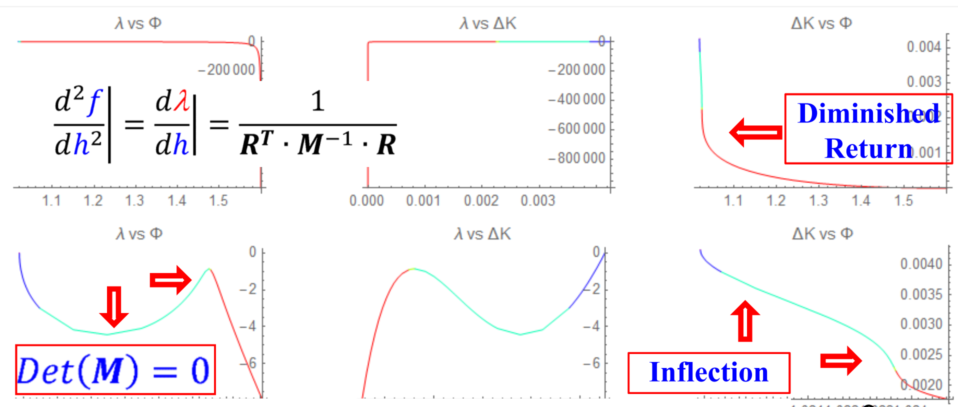
More Advantages

- Works on any system, including XY-coupled and interspersed modules
- Computational demand is a <u>slow</u> function of optics/system complexity
- Systematic procedure to map out and isolate <u>Global</u> optimum (Pareto Front)
- Complete range of options for optimal trade-off (Ideal for <u>distributed matching</u>)
- Points of diminished return identified by well-defined procedure (Det(M)=0)
- Not dealing with a black box
- Determinism, Robustness and Reproducibility are important for feedback applications



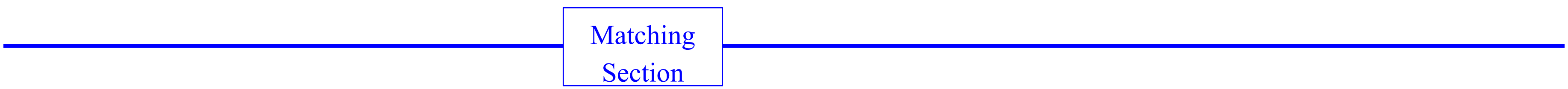
Try Solving $\nabla\Phi = 0$ for 5 Quads



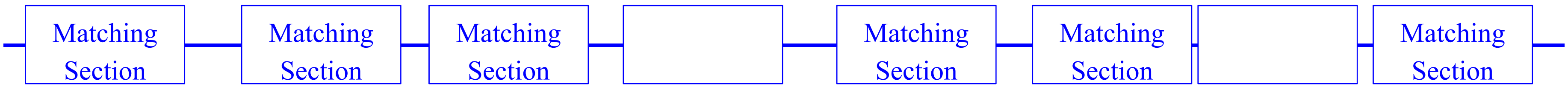


Each One A Serious Challenge to Conventional Methods

Implementing Distributed Matching Profile and Transport

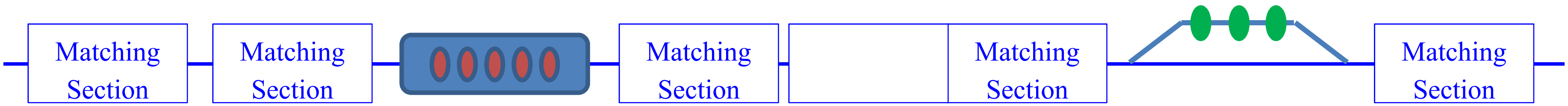


Implementing Distributed Matching Profile and Transport



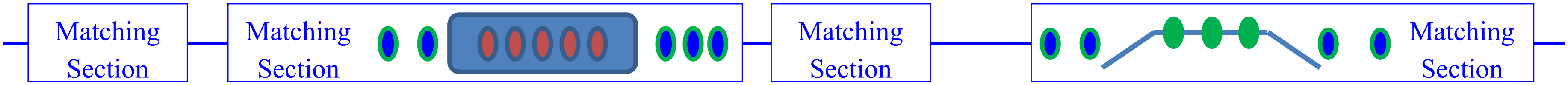
- Subdivide line into matching sections

Implementing Distributed Matching Profile and Transport



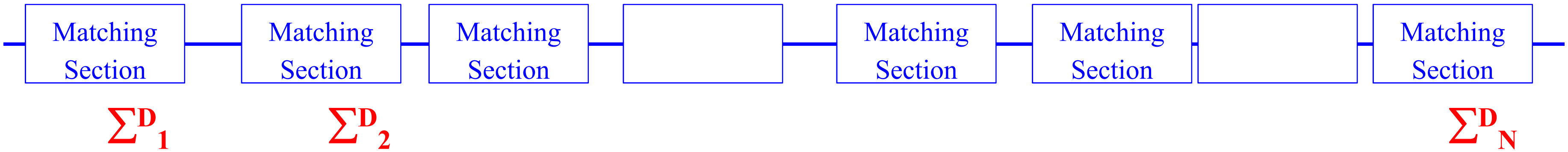
- ❖ Subdivide line into matching sections

Implementing Distributed Matching Profile and Transport



❖ Subdivide line into matching sections

Implementing Distributed Matching Profile and Transport

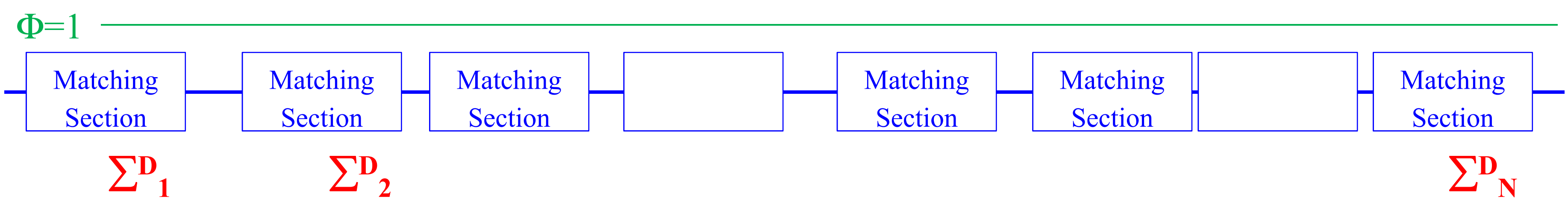


- ❖ Subdivide line into matching sections
- ➢ Matching target for each section is **Fixed**

Implementing Distributed Matching Profile and Transport

❖ To Fix Beam Profile Mismatch

Φ_1

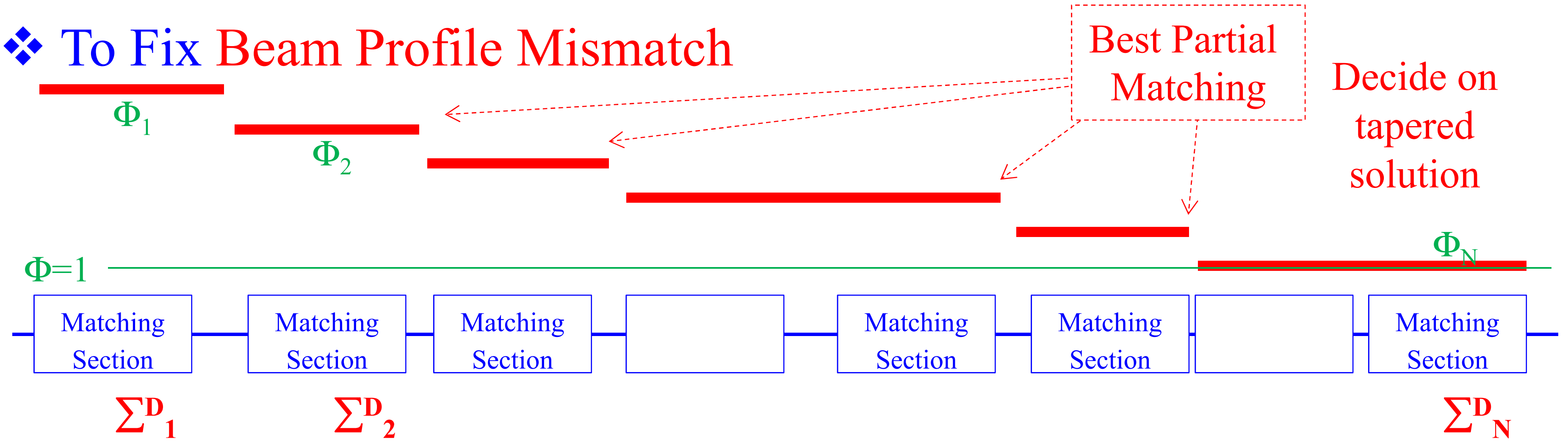


❖ Subdivide line into matching sections

➢ Matching target for each section is **Fixed**

Implementing Distributed Matching Profile and Transport

❖ To Fix Beam Profile Mismatch

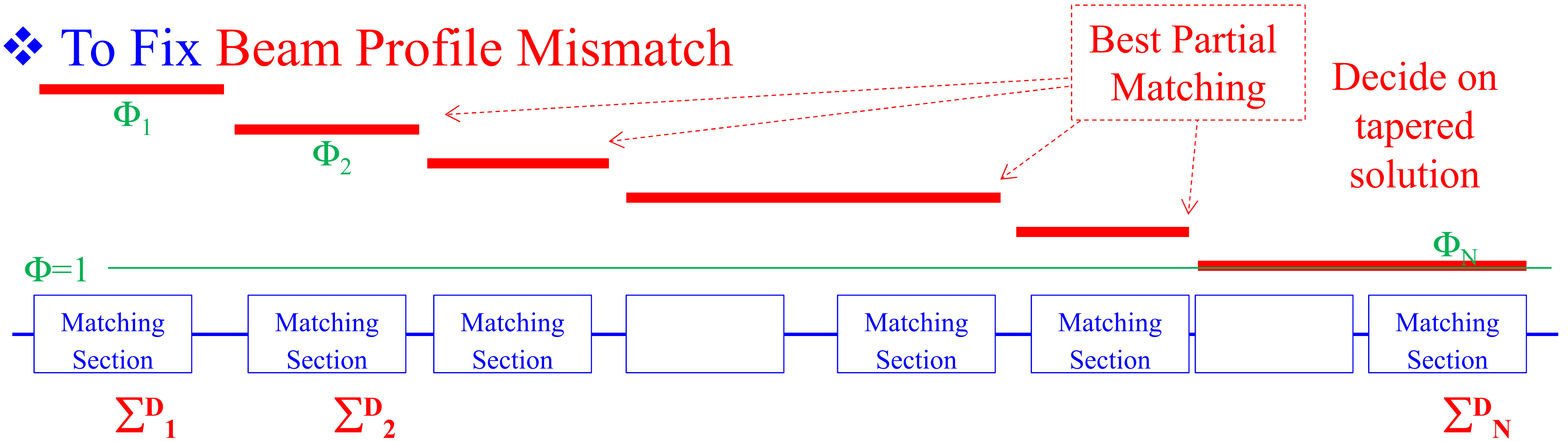


❖ Subdivide line into matching sections

➢ Matching target for each section is **Fixed**

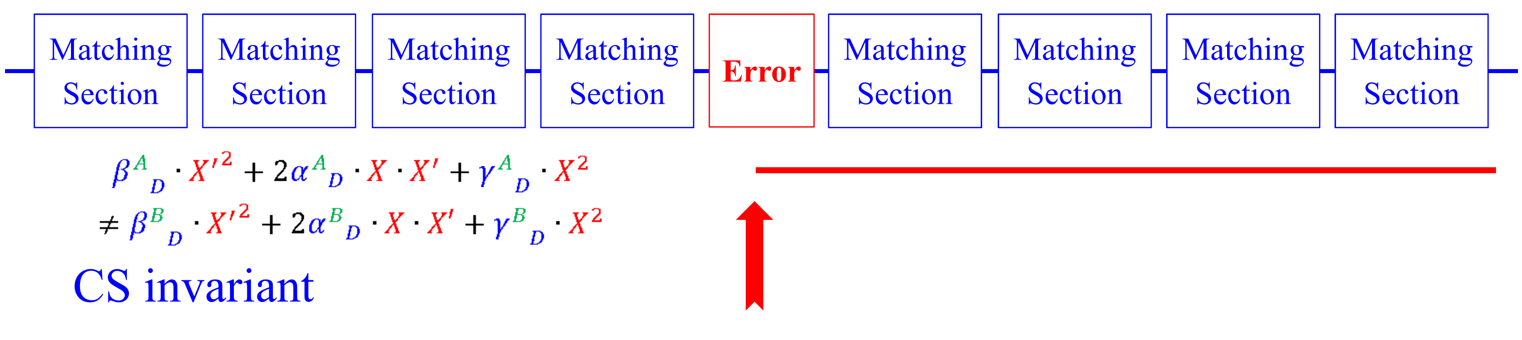
Implementing Distributed Matching Profile and Transport

❖ To Fix Beam Profile Mismatch



- ❖ Subdivide line into matching sections
  - ➢ Matching target for each section is **Fixed**

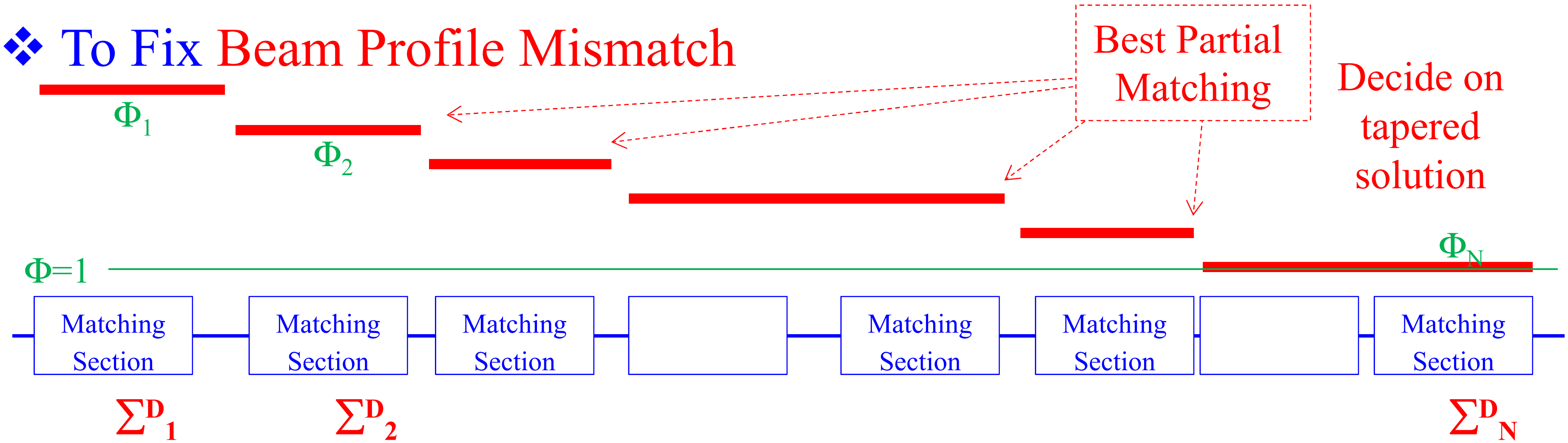
❖ To Fix Optics/Transport Error



$$\beta^{A}{}_{D} \cdot X'^{2} + 2\alpha^{A}{}_{D} \cdot X \cdot X' + \gamma^{A}{}_{D} \cdot X^{2} \neq \beta^{B}{}_{D} \cdot X'^{2} + 2\alpha^{B}{}_{D} \cdot X \cdot X' + \gamma^{B}{}_{D} \cdot X^{2}$$

Implementing Distributed Matching Profile and Transport

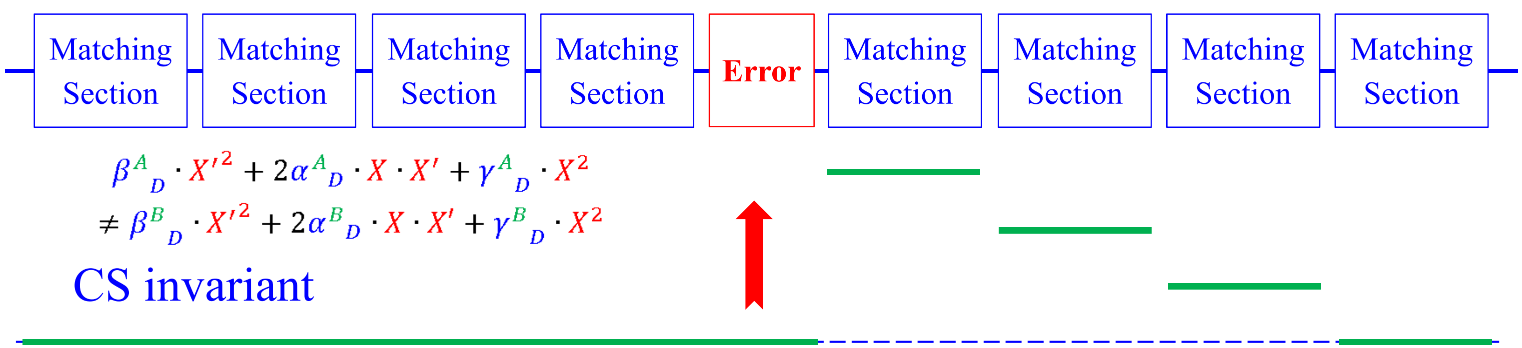
❖ To Fix Beam Profile Mismatch



❖ Subdivide line into matching sections

➢ Matching target for each section is **Fixed**

❖ To Fix Optics/Transport Error



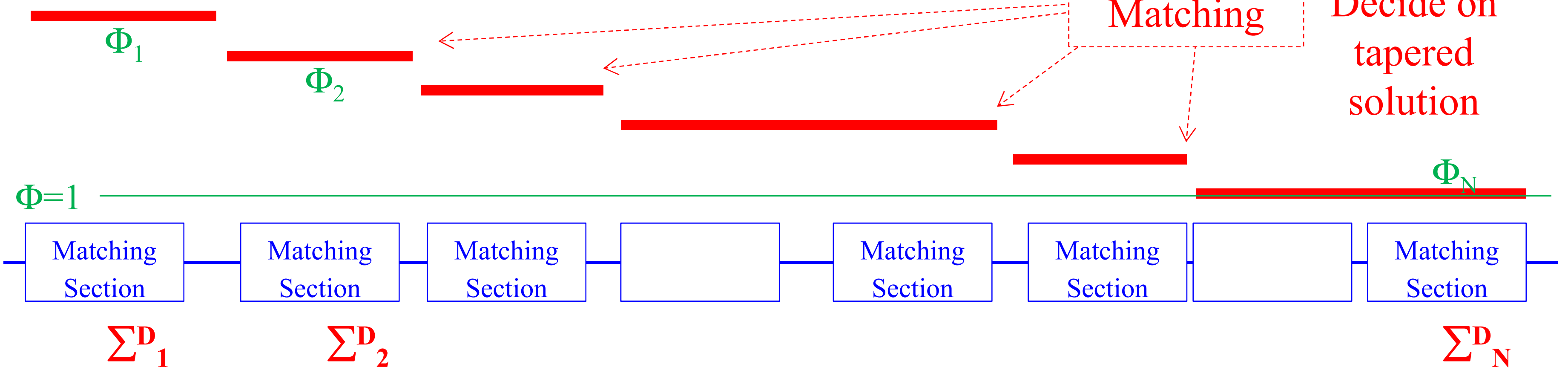
$$\beta^A{}_D \cdot X'^2 + 2\alpha^A{}_D \cdot X \cdot X' + \gamma^A{}_D \cdot X^2 \neq \beta^B{}_D \cdot X'^2 + 2\alpha^B{}_D \cdot X \cdot X' + \gamma^B{}_D \cdot X^2$$

CS invariant

Match after error

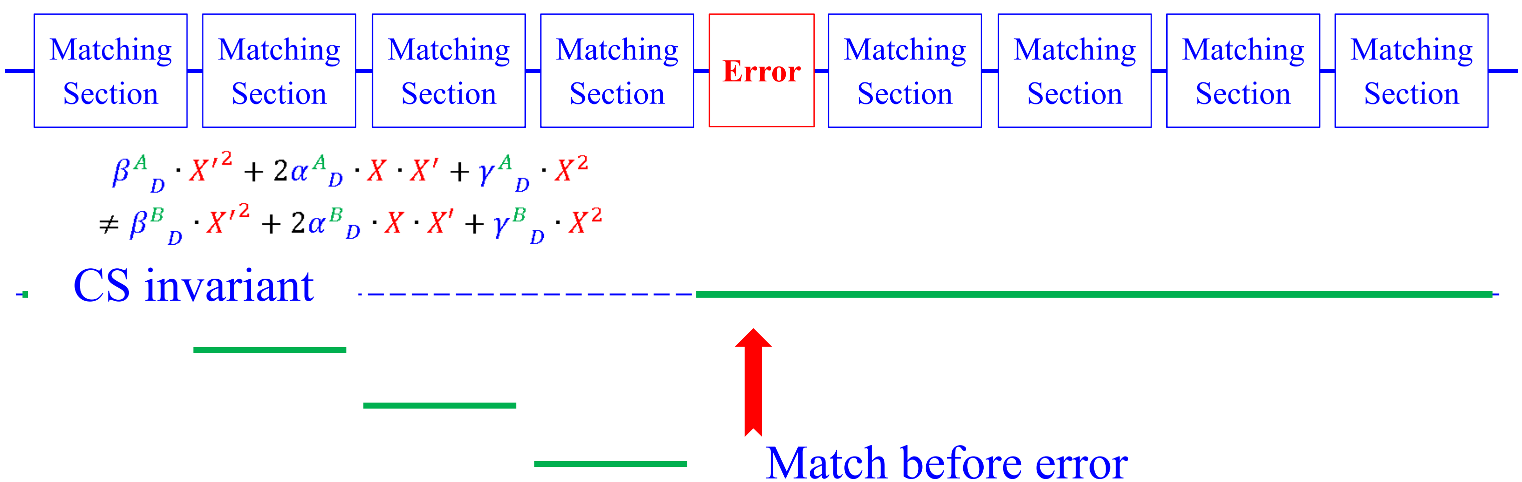
Implementing Distributed Matching Profile and Transport

❖ To Fix Beam Profile Mismatch



- ❖ Subdivide line into matching sections
- ➢ Matching target for each section is **Fixed**

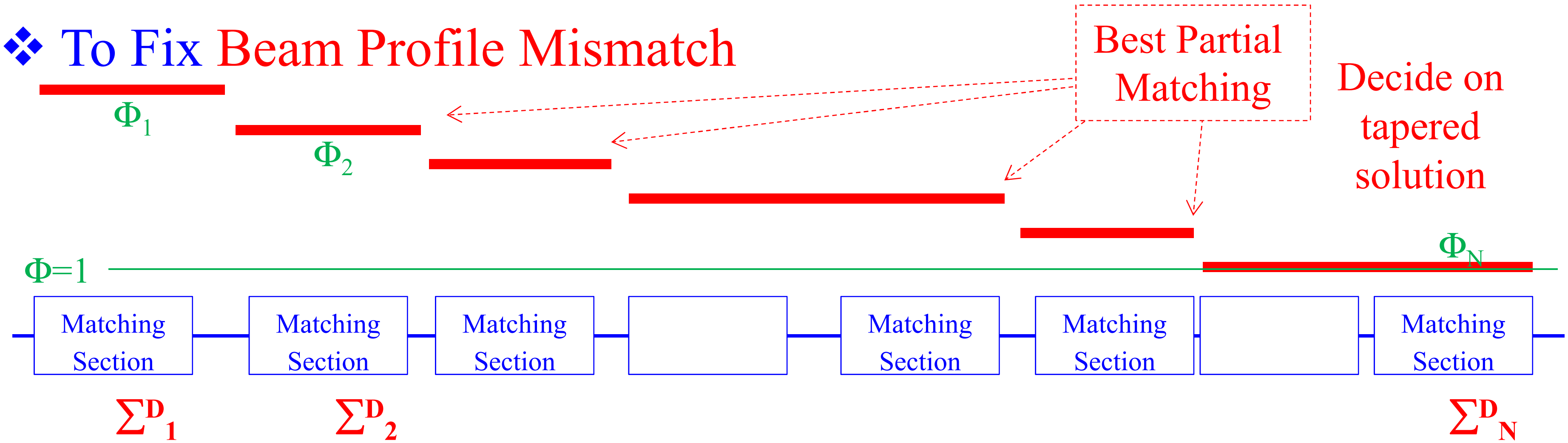
❖ To Fix Optics/Transport Error



$$\beta^A{}_D \cdot X'^2 + 2\alpha^A{}_D \cdot X \cdot X' + \gamma^A{}_D \cdot X^2 \neq \beta^B{}_D \cdot X'^2 + 2\alpha^B{}_D \cdot X \cdot X' + \gamma^B{}_D \cdot X^2$$

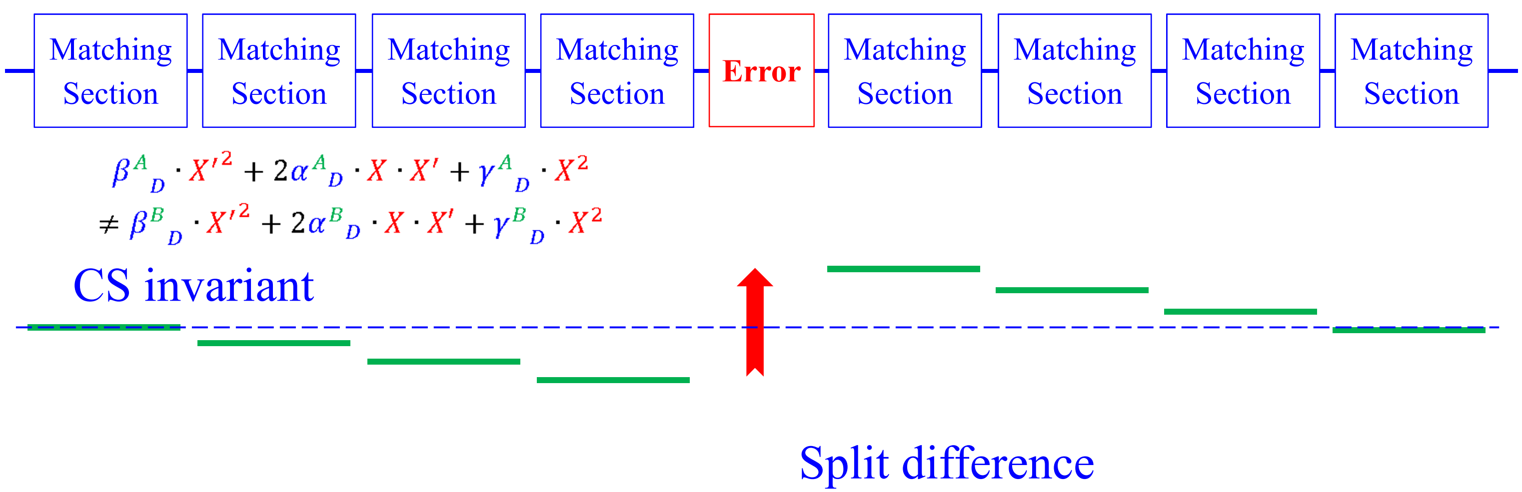
Implementing Distributed Matching Profile and Transport

❖ To Fix Beam Profile Mismatch



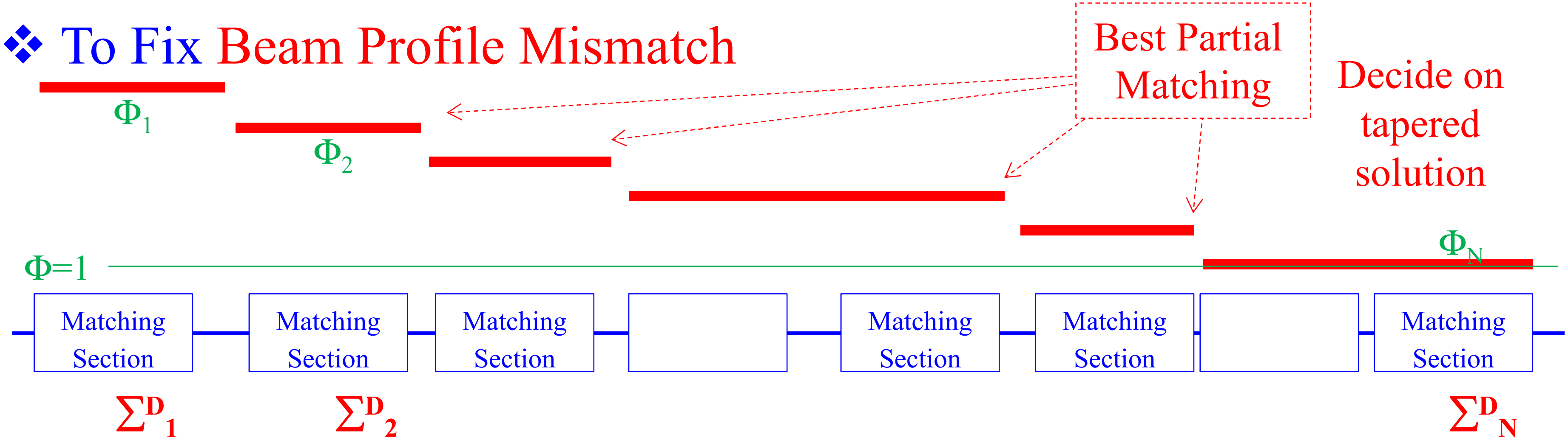
- ❖ Subdivide line into matching sections
- ➢ Matching target for each section is **Fixed**

❖ To Fix Optics/Transport Error



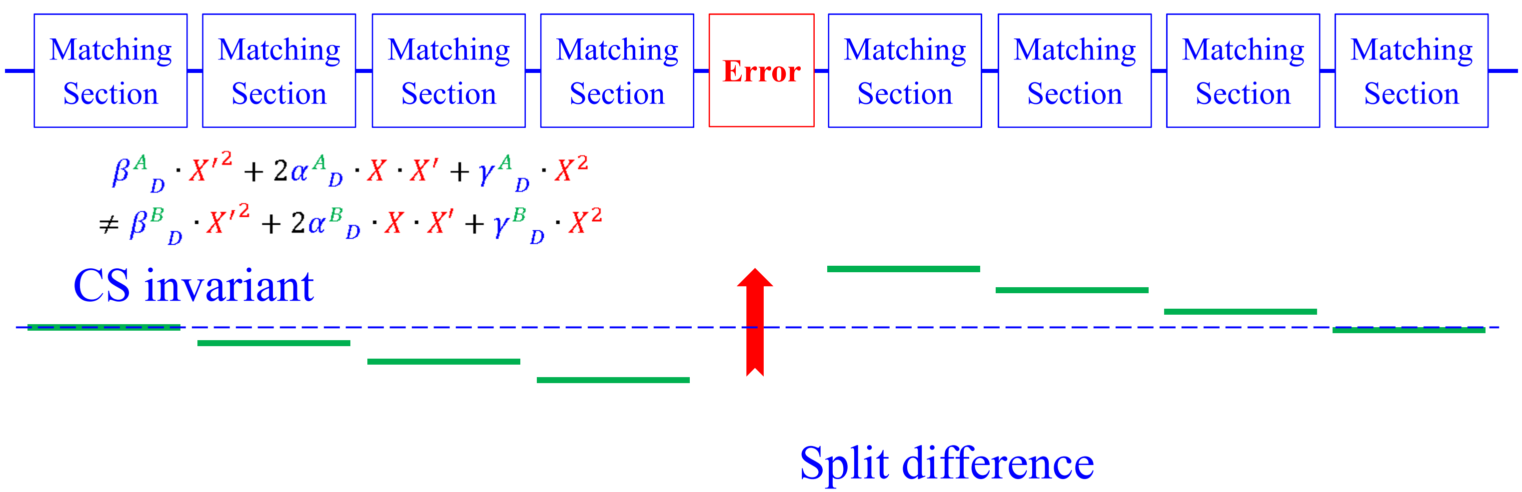
Implementing Distributed Matching Profile and Transport

- To Fix Beam Profile Mismatch



- Subdivide line into matching sections
  - Matching target for each section is **Fixed**

- To Fix Optics/Transport Error



User has freedom on solution scenario, e.g. How to taper mismatch profile

Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During Online operation simply interpolate from Offline results.
  - Speed & Predictability

Example (3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

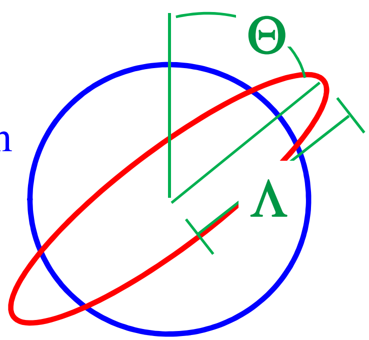
$$\Phi_{X/Y} = 9$$

$$\Theta_{X/Y} = 0 \rightarrow \pi$$

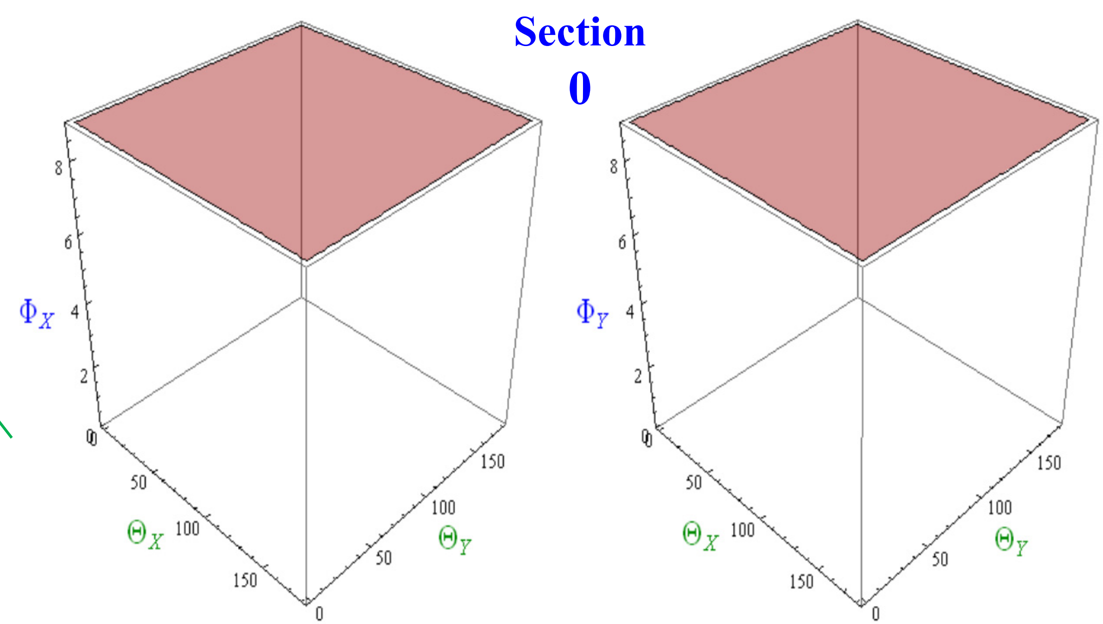
Normalized Design Beam

Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching



Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During Online operation simply interpolate from Offline results.
  - Speed & Predictability

Example (3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

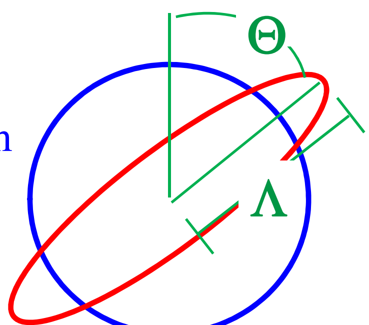
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- Interpolate for partial matching in subsequent sections.

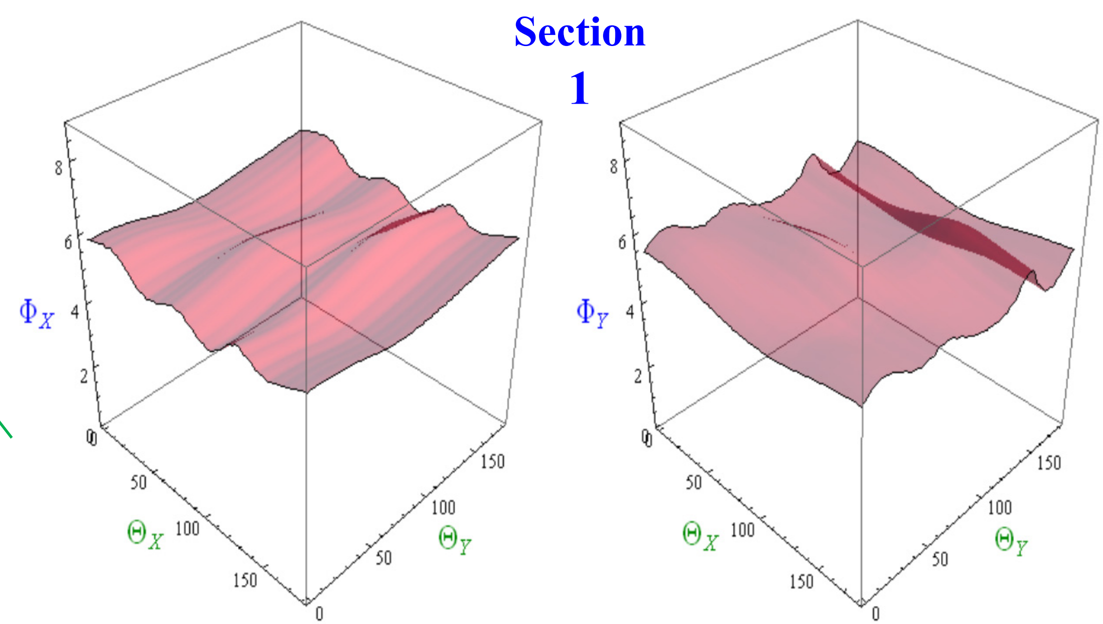
Normalized Design Beam

Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching



Why Perform Matching on Beam Time? – (3rd) Alternate View

- ❖ Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- ❖ As functions of input mismatch and embedded modules (e.g. RF phase)
- ❖ During Online operation simply interpolate from Offline results.
  - ➢ Speed & Predictability

Example (3-quad section 120° FODO):

- ❖ Construct interpolation table covering range:
  - ➢ Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - ➢ Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- ❖ Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

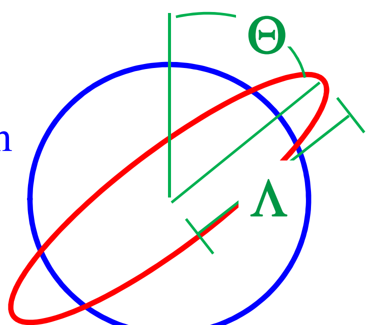
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- ❖ Interpolate for partial matching in subsequent sections.

Normalized Design Beam

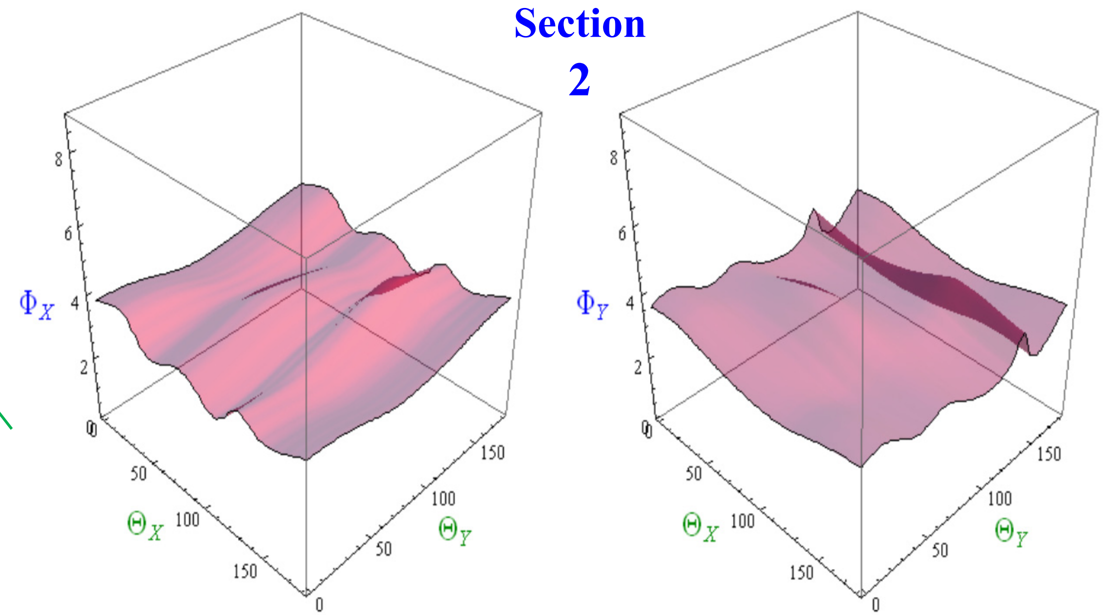
Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching

Section 2



Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During <u>Online</u> operation simply interpolate from <u>Offline</u> results.
  - Speed & Predictability

<u>Example (</u>3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

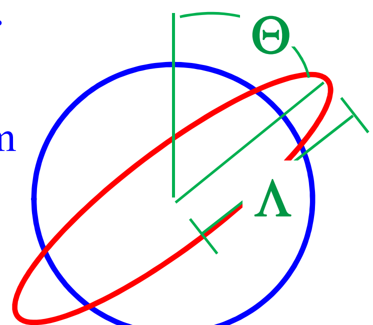
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- Interpolate for <u>partial</u> matching in subsequent sections.

Normalized Design Beam

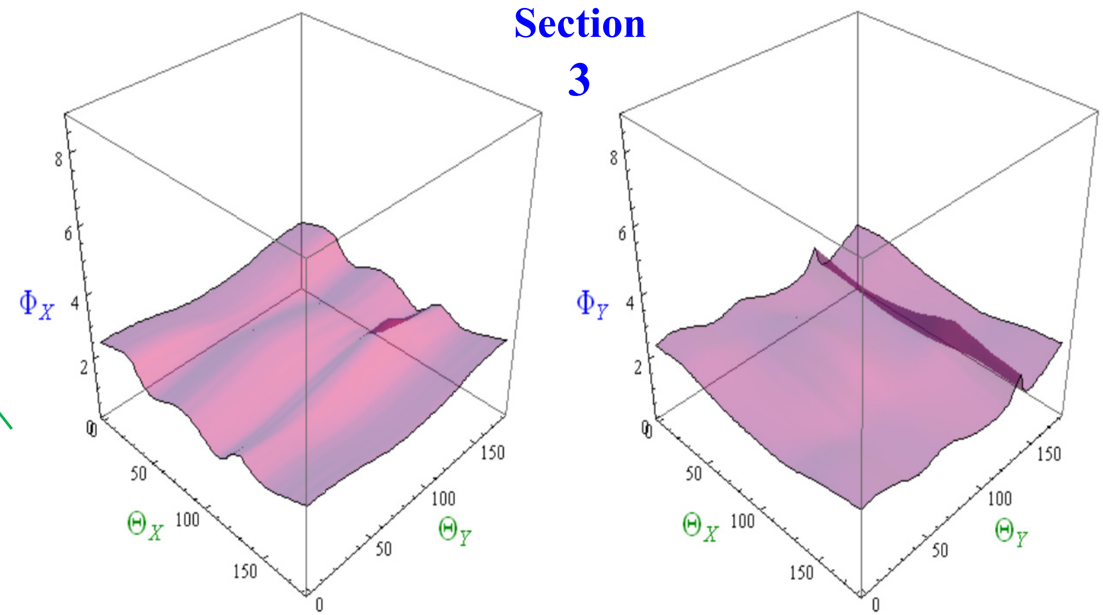
Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching

Section 3



Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During Online operation simply interpolate from Offline results.
  - Speed & Predictability

Example (3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

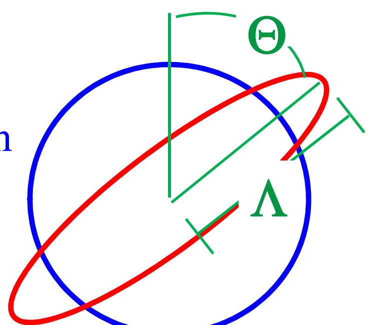
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- Interpolate for partial matching in subsequent sections.

Normalized Design Beam

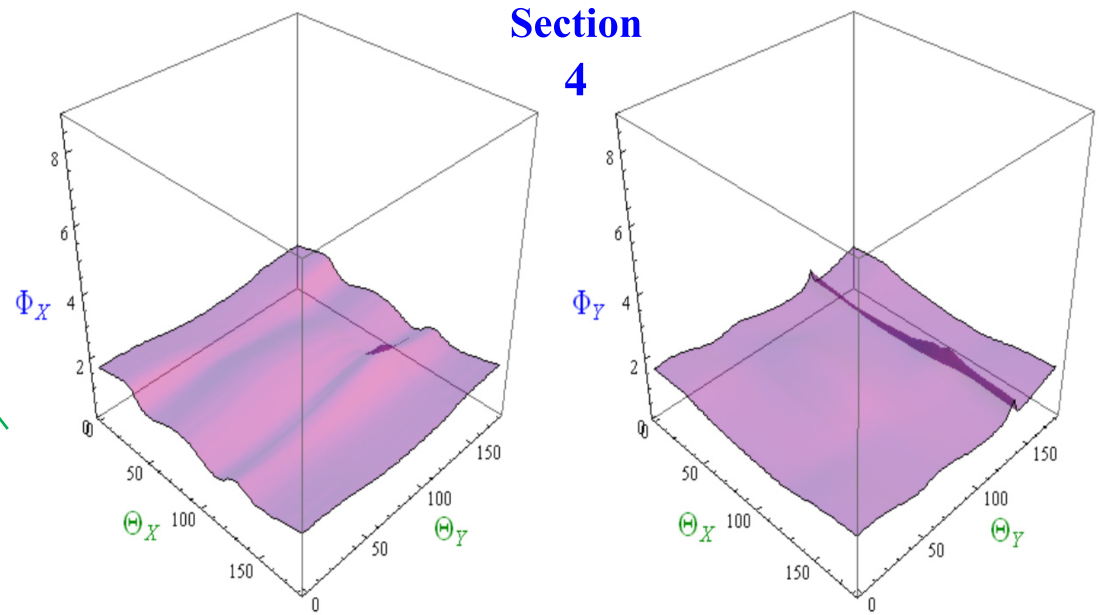
Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching

Section 4



Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During Online operation simply interpolate from Offline results.
  - Speed & Predictability

Example (3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

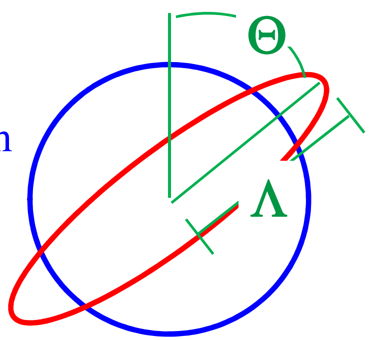
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- Interpolate for partial matching in subsequent sections.

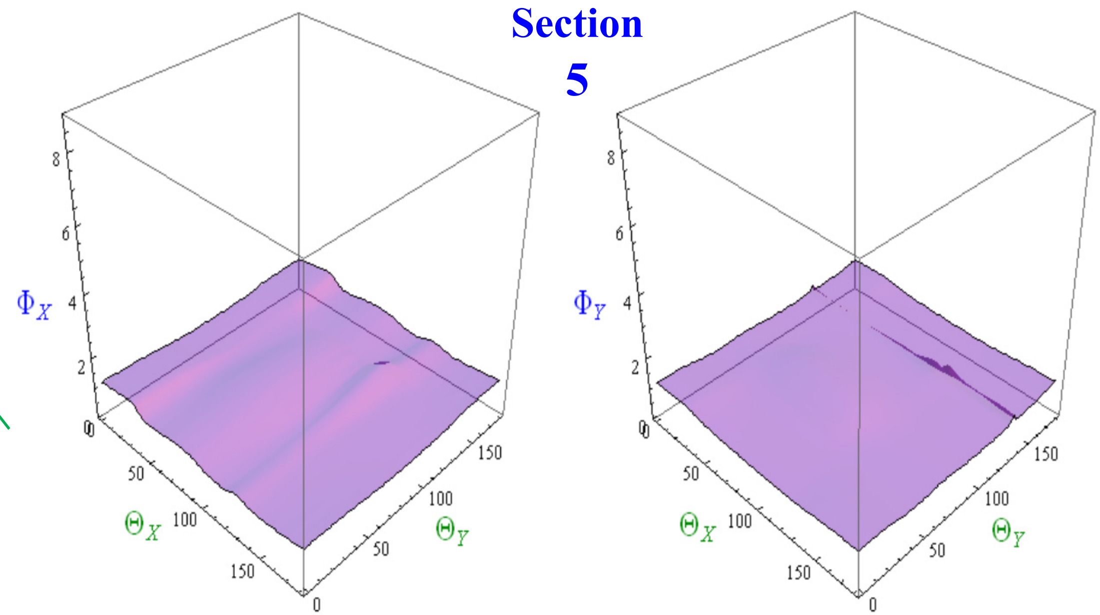
Normalized Design Beam

Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching



Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During Online operation simply interpolate from Offline results.
  - Speed & Predictability

Example (3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

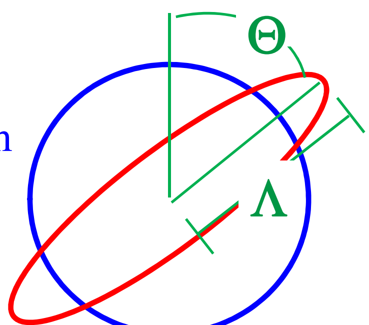
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- Interpolate for partial matching in subsequent sections.

Normalized Design Beam

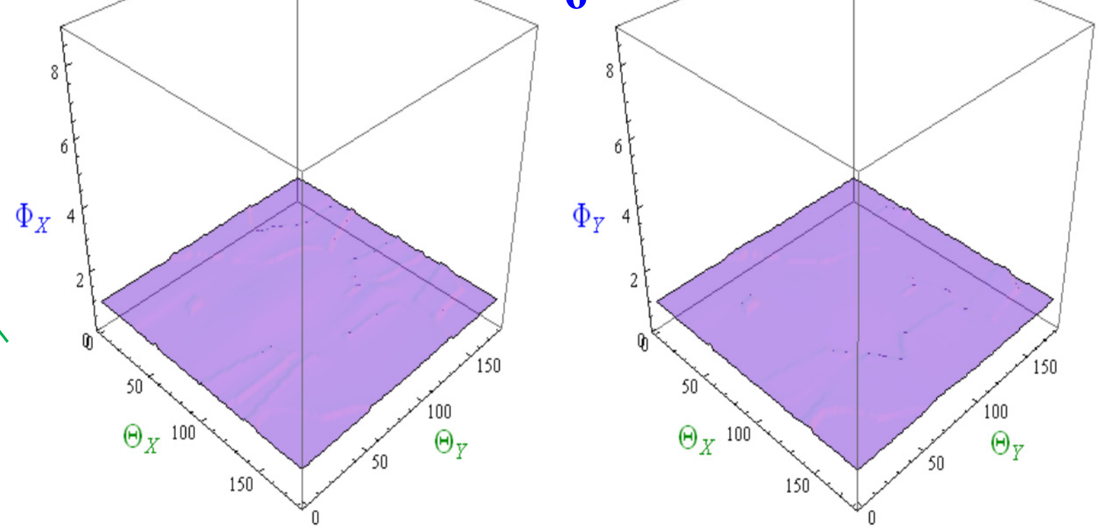
Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching

Section 6



Why Perform Matching on Beam Time? – (3rd) Alternate View

- ❖ Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- ❖ As functions of input mismatch and embedded modules (e.g. RF phase)
- ❖ During Online operation simply interpolate from Offline results.
  - ➢ Speed & Predictability

Example (3-quad section 120° FODO):

- ❖ Construct interpolation table covering range:
  - ➢ Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - ➢ Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- ❖ Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

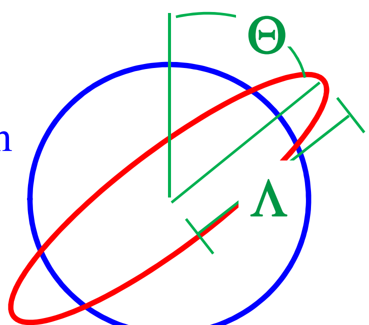
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- ❖ Interpolate for partial matching in subsequent sections.

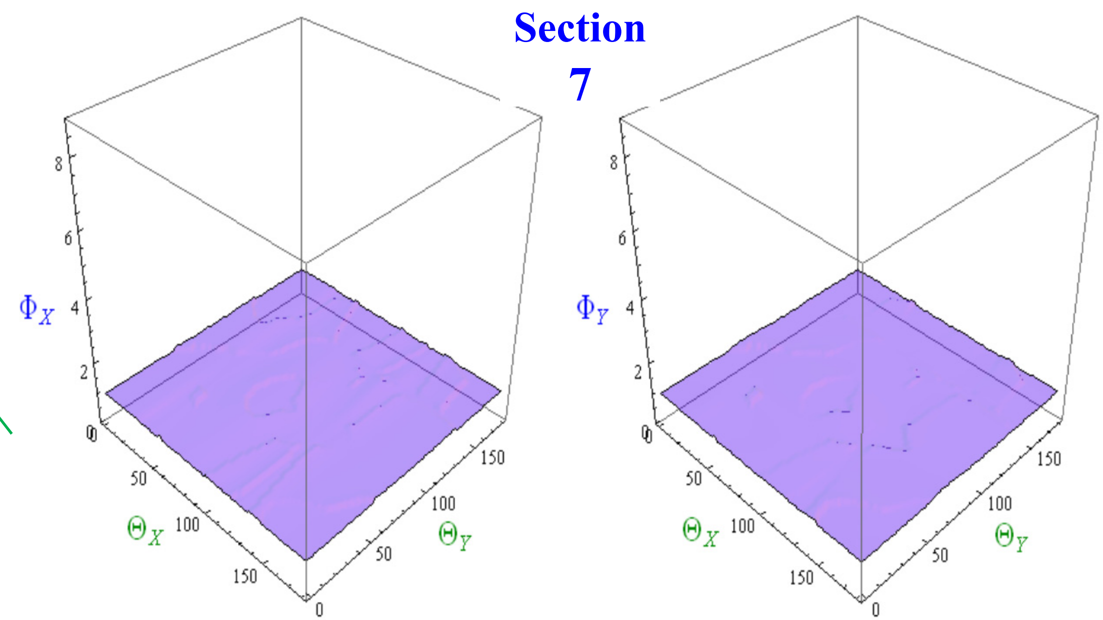
Normalized Design Beam

Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Evolution of beam through successive matching



Why Perform Matching on Beam Time? – (3rd) Alternate View

- Matching targets are fixed ⇒ Pre-compute trade-off solutions Offline
- As functions of input mismatch and embedded modules (e.g. RF phase)
- During Online operation simply interpolate from Offline results.
  - Speed & Predictability

Example (3-quad section 120° FODO):

- Construct interpolation table covering range:
  - Input Mismatch Amp. $\Phi_{X/Y} = 1 \rightarrow 9$
  - Input Mismatch Angle $\Theta_{X/Y} = 0 \rightarrow \pi$
- Launch beam with initial mismatch:

$$\Phi_{X/Y} = 9$$

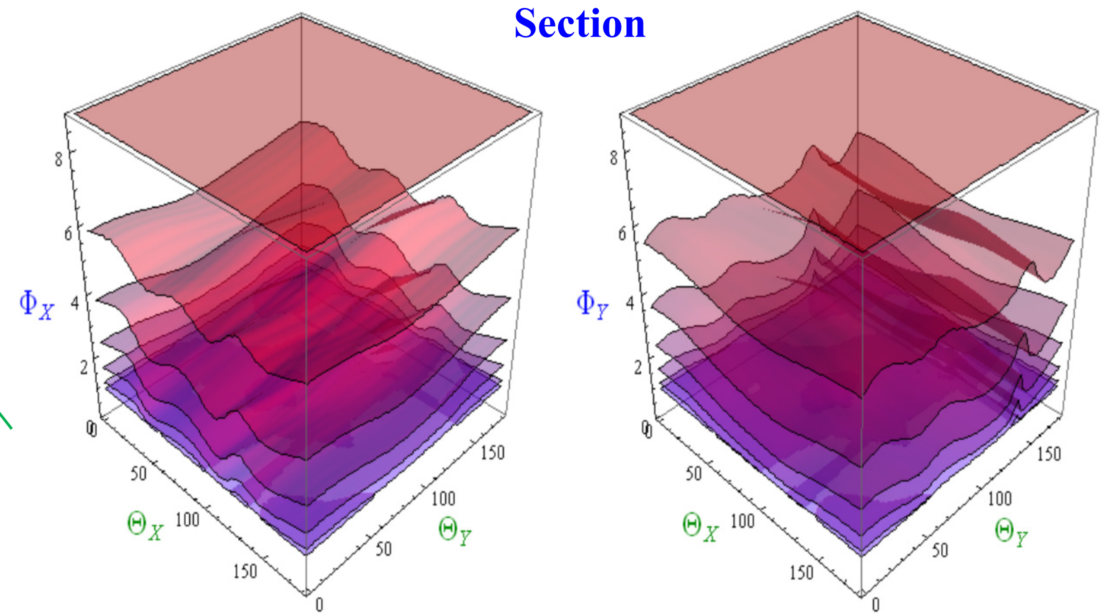
$$\Theta_{X/Y} = 0 \rightarrow \pi$$

- Interpolate for partial matching in subsequent sections.

Determinism in algorithm is crucial to generating Massive interpolation tables!

Evolution of beam through successive matching

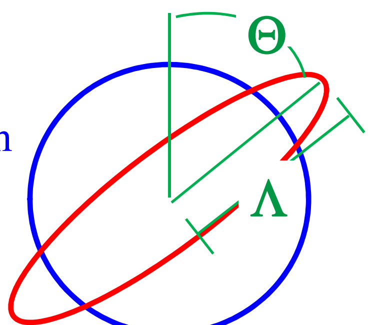
Section



Normalized Design Beam

Mismatched Beam

$$\Lambda = \sqrt{\Phi + \sqrt{\Phi^2 - 1}}$$



Recap

- Possibility to Realize 3 Alternate Views to Matching
  - Distributed instead of Local
  - Optimizing Tradeoff Deterministically instead of Single objective
  - Offline Computation instead of Online
- Interlinked Concepts But Do Not Require Monolithic Implementation
  - Enabling component is stand-alone Matching Engine.
  - Distributed Scheme
  - Interpolated Solution
- Application Beyond Matching
  - Works on any parameter with well-behaved analytic model
  - Determinism can be maintained even when starting point is not known a priori

  ⇒ (Impose Artificial Constraint)

Input / Idea of Application Welcome!

New Optimization Algorithm
Rigor
Determinism
Extensive adaptability
Flexibility
Insight

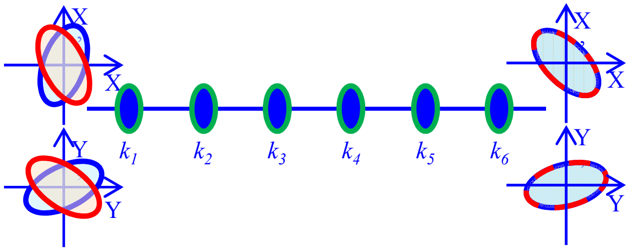
Solution by Interpolation
Speed
Predictability
Reproducibility
Control flexibility
Insight

Distributed Matching Configuration
Smooth envelope/jitter
Reduced compounded error
Global matching improvement
Recourse to matching error
Low quad strength
Flexibility

Best Taken into Account at Design Stage

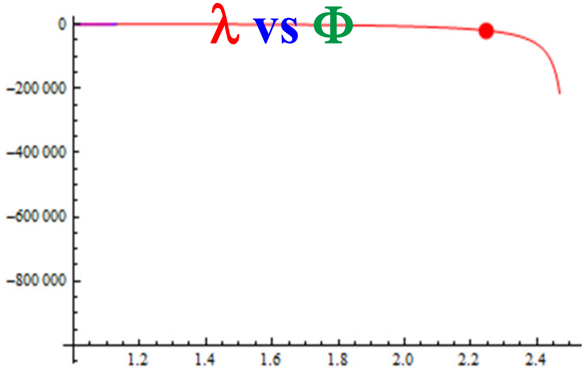
If λ ≠0, Don't Stop

- 30° FODO; 6 Quad Matching;

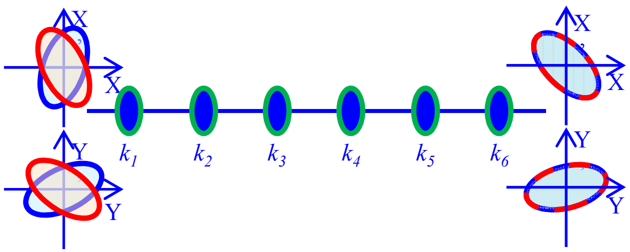


Evolution

λ vs Φ



If λ ≠0, Don't Stop

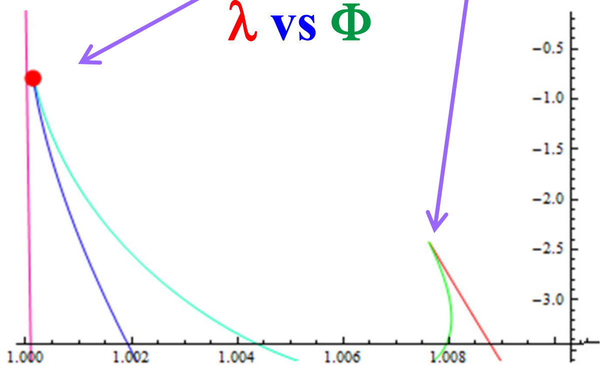


- 30° FODO; 6 Quad Matching;

Φ	1.00013	1.0076
λ	-0.74	-2.42

Evolution

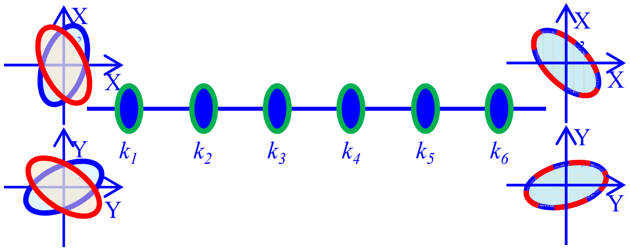
λ vs Φ



If λ ≠0, Don't Stop

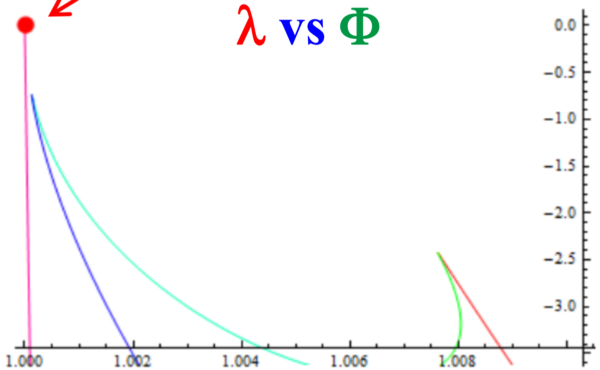
- ❖ 30° FODO; 6 Quad Matching;

Φ	1	1.00013	1.0076
λ	0	-0.74	-2.42



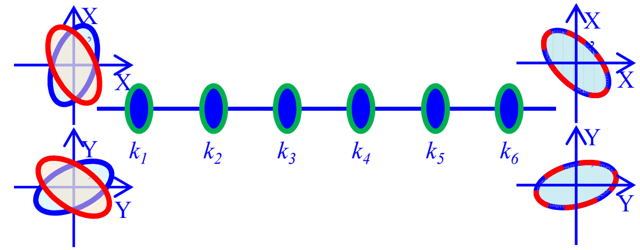
Evolution

λ vs Φ



If λ ≠0, Don't Stop

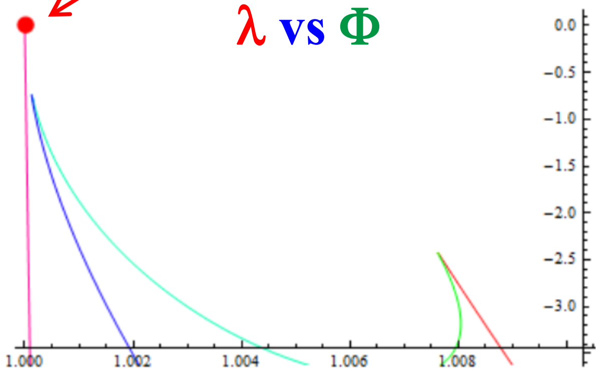
❖ 30° FODO; 6 Quad Matching;



Φ	1	1.00013	1.0076
λ	0	-0.74	-2.42

Evolution

λ vs Φ

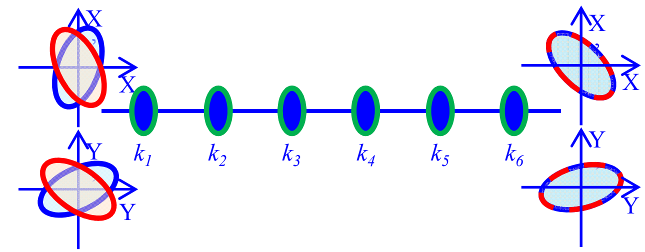


Why Bother with 10^{-4} ?

If λ ≠0, Don't Stop

❖ 30° FODO; 6 Quad Matching;

Φ	1	1.00013	1.0076
λ	0	-0.74	-2.42

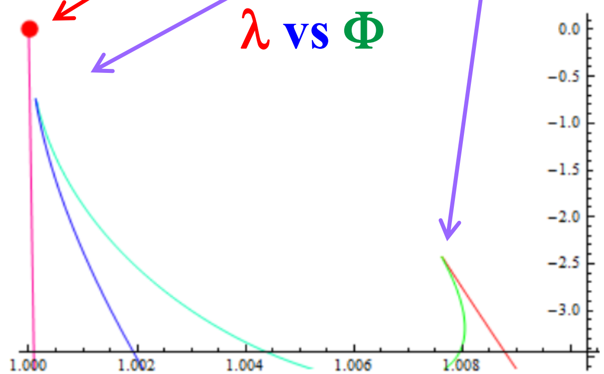


ΔK Gain of > 50%

By insisting on λ→0

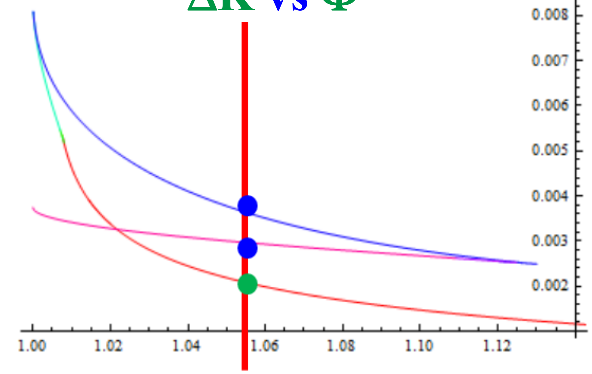
Evolution

λ vs Φ



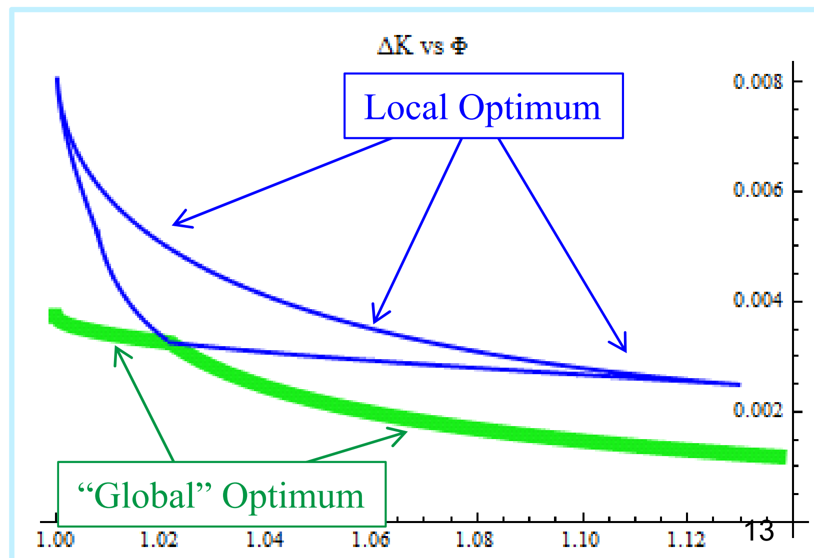
Φ	λ	ΔK
1.00013	-0.74	0.008
1.0076	-2.42	0.005
1	0	0.003

ΔK vs Φ



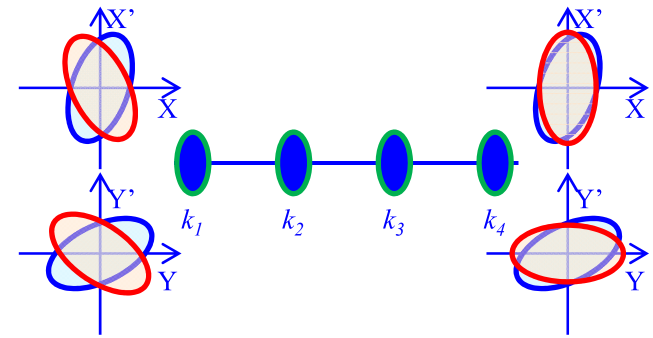
Entire Path ⇒ Local vs Global Optimum

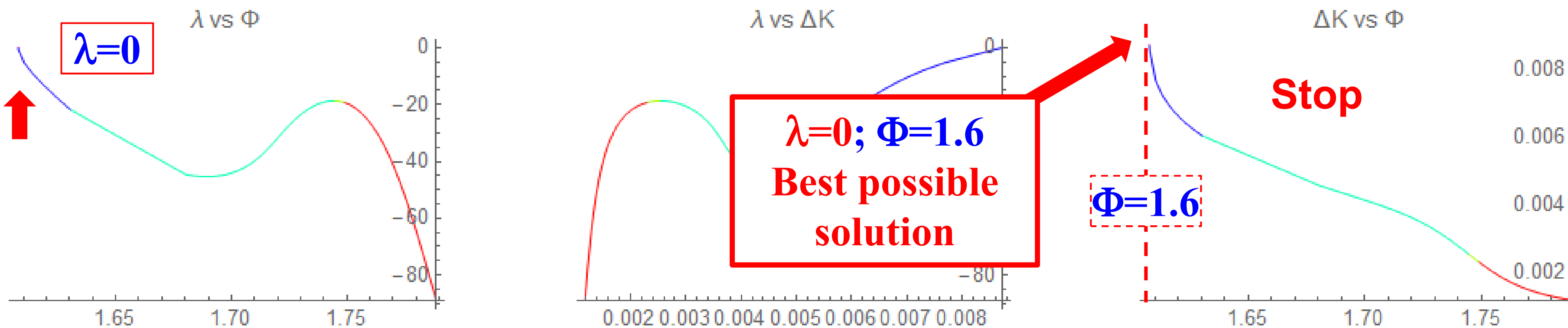
- ❖ Local optimal condition is satisfied every-where, but only some are "Global".
- ❖ Isolate **global** optima by short-circuiting inferior **local** optima.
  - ➢ Green curve is always monotonic ($\lambda < 0$)
- ❖ Akin to "Pareto Front" concept in multi-objective optimization



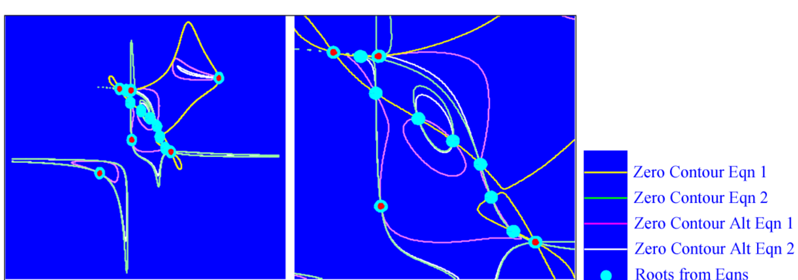
If λ=0, Stop

- 120° FODO; 4 Quad Matching Section;

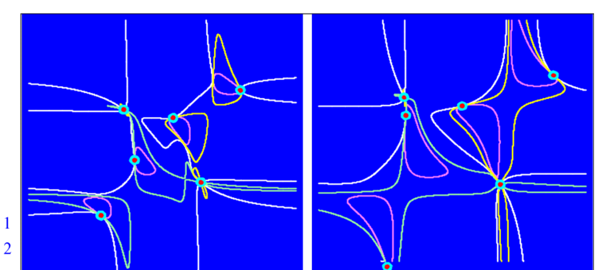




- This 4-Quad Mismatched Configuration Does Not Allow 100% Matching (All Roots Are Complex).
- Conventional Algorithm Cannot Give Unequivocal Answer Like This.



Example of Real Roots for a Well-Configured System.

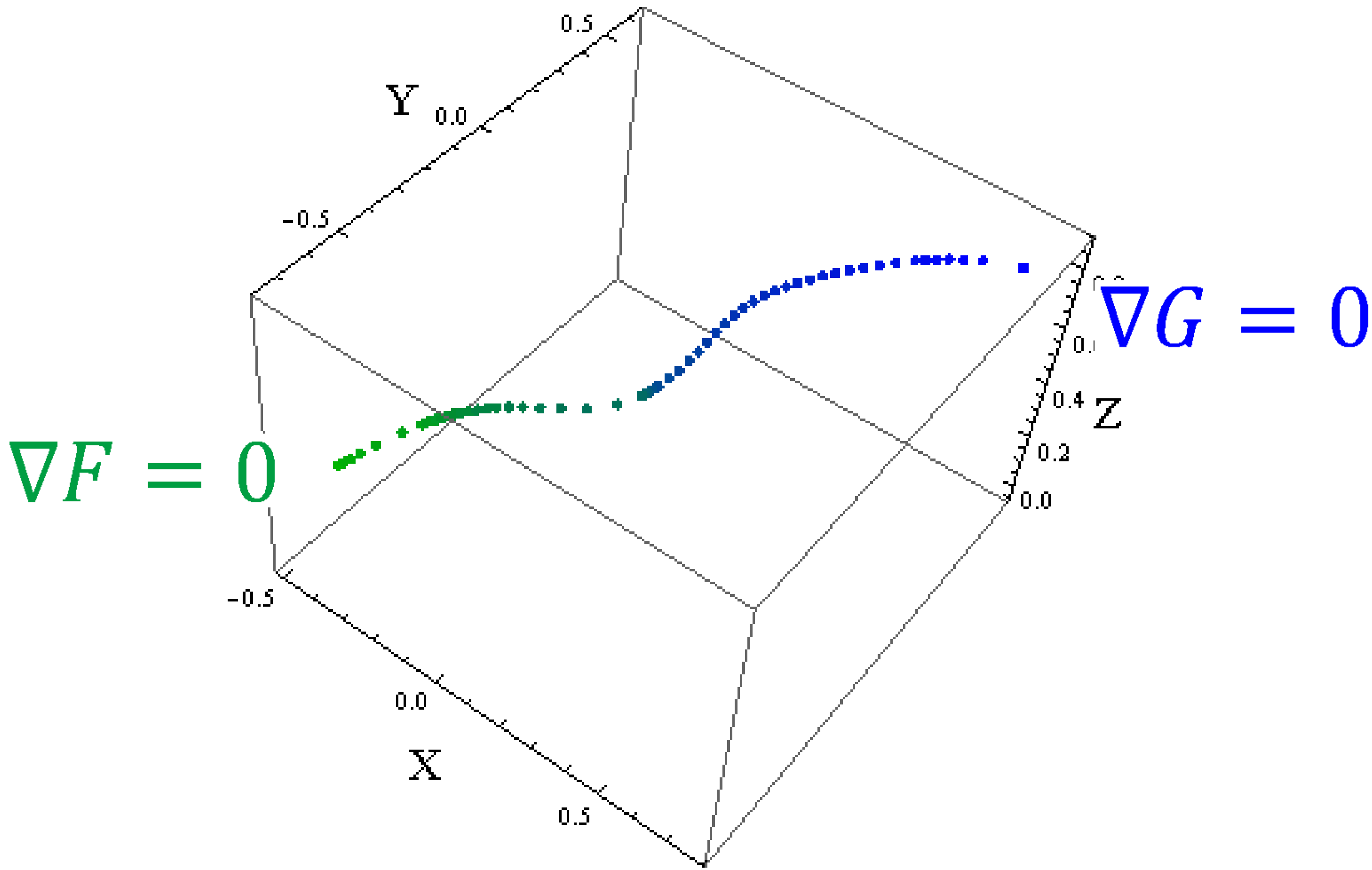


Other Examples of Complete Absence of Real Roots.

4-Quad Matching with
No Real Roots
Y. Chao PAC 2001

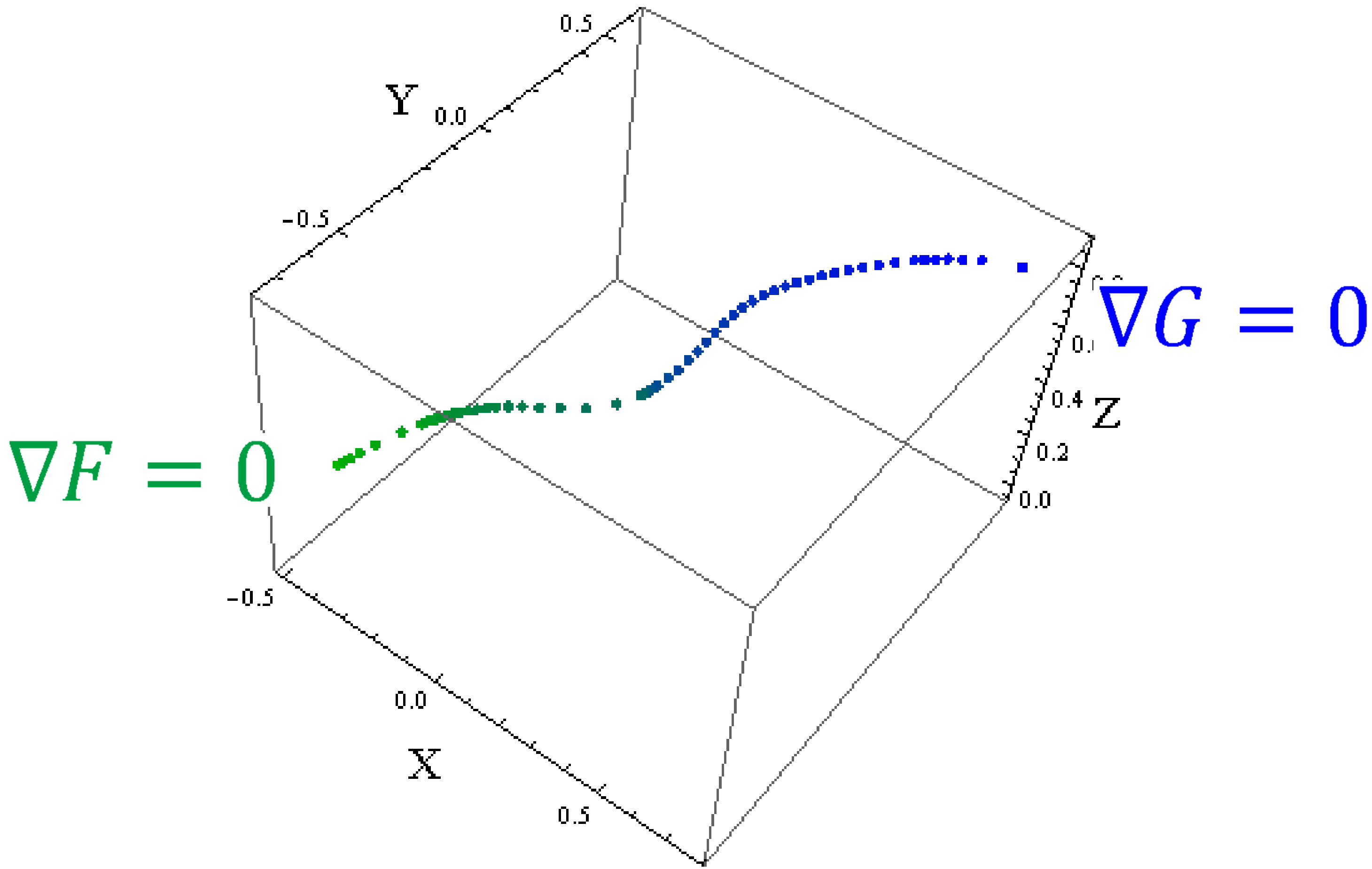
Application beyond Matching? – Restoring Determinism

- Algorithm should work on any other function with an analytical model.
- Determinism depends on "known" starting point. ($\nabla H=0$, $\Delta k_m=0$ or $k_m=0$)
- What if neither $\nabla F=0$ nor $\nabla G=0$ is known a priori? Determinism Lost?



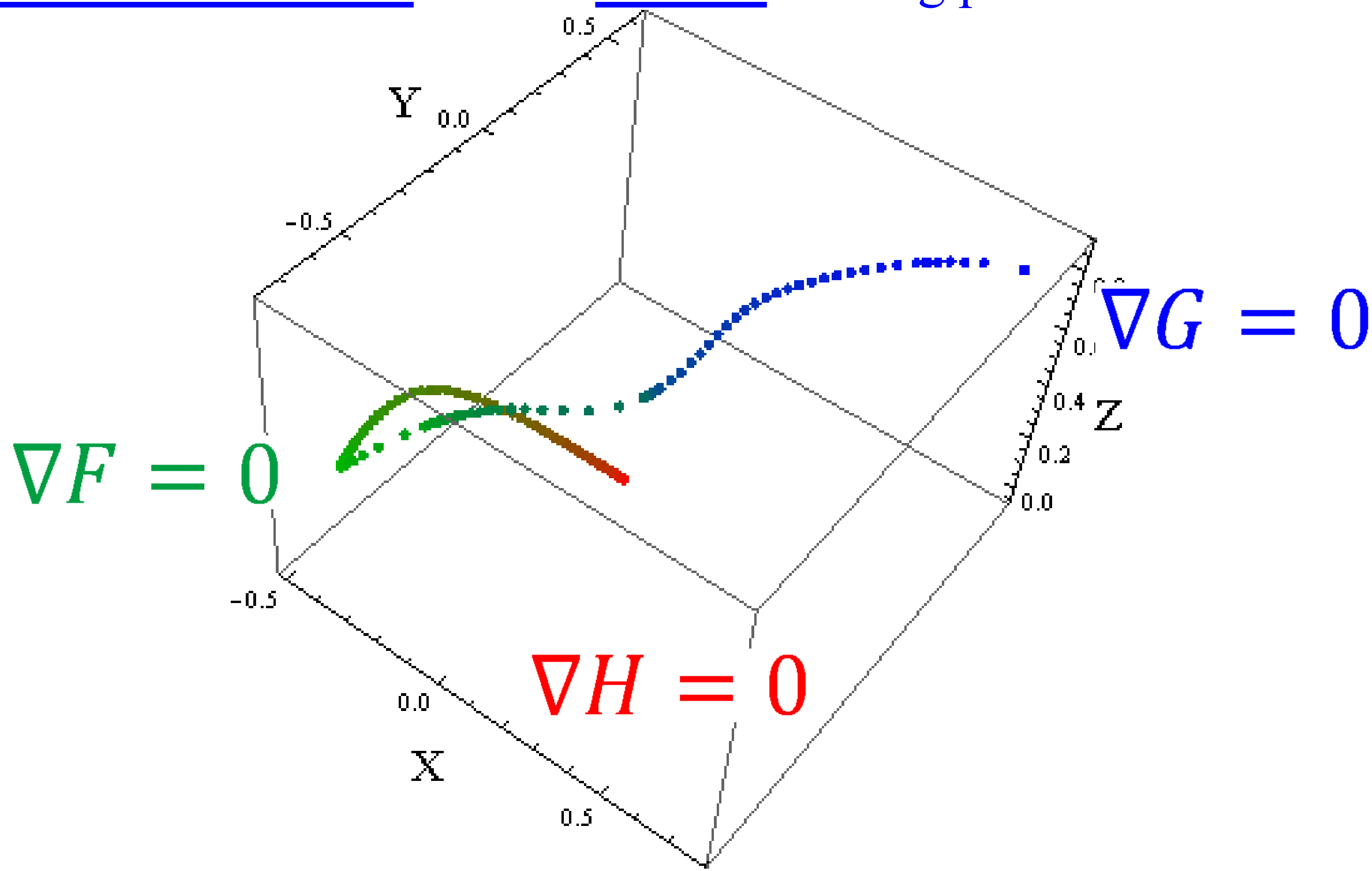
Application beyond Matching? – Restoring Determinism

- ❖ Algorithm should work on any other function with an analytical model.
- ❖ Determinism depends on "known" starting point. ($\nabla \boldsymbol{H}$=0, $\Delta \boldsymbol{k_m}$=0 or $\boldsymbol{k_m}$=0)
- ❖ What if neither $\nabla \boldsymbol{F}$=0 nor $\nabla \boldsymbol{G}$=0 is known a priori? Determinism Lost?
  - ➢ Note $\nabla \boldsymbol{F}$=0 is a common terminus to trade-off with all other constraints.



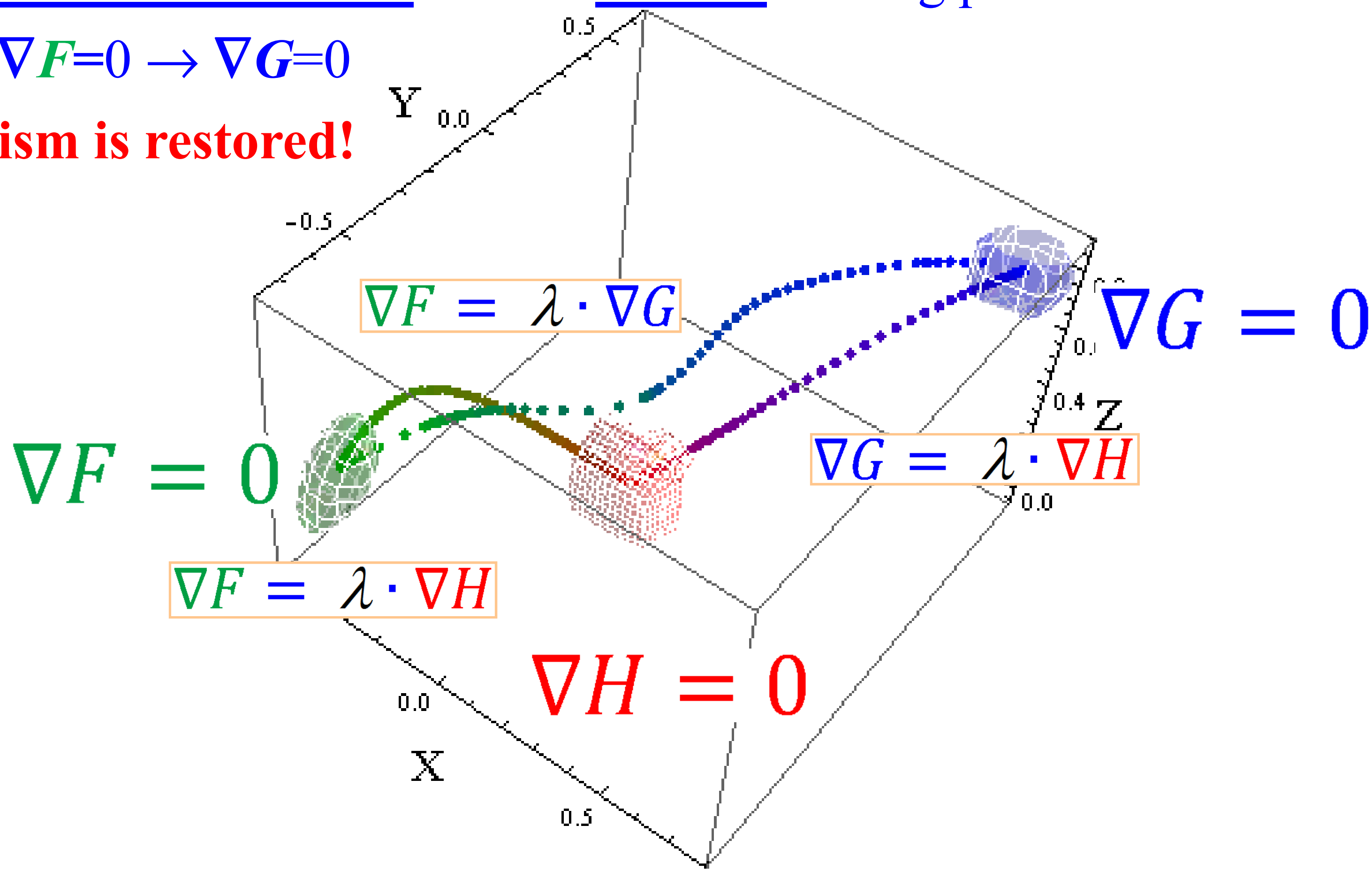
Application beyond Matching? – Restoring Determinism

- Algorithm should work on any other function with an analytical model.
- Determinism depends on "known" starting point. (∇H=0, Δk_m=0 or k_m=0)
- What if neither ∇F=0 nor ∇G=0 is known a priori? Determinism Lost?
  - Note ∇F=0 is a common terminus to trade-off with all other constraints.
  - Create **Artificial Constraint** H with **Known** starting point ∇H=0



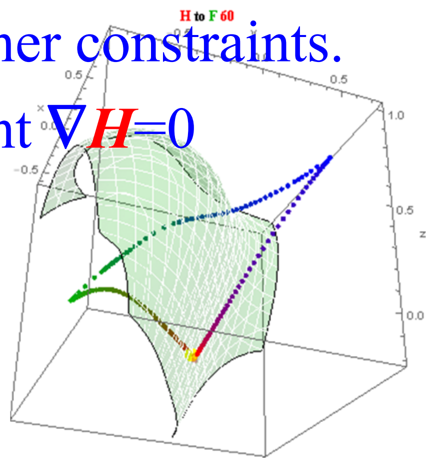
Application beyond Matching? – Restoring Determinism

- Algorithm should work on any other function with an analytical model.
- Determinism depends on "known" starting point. (∇H=0, Δk_m=0 or k_m=0)
- What if neither ∇F=0 nor ∇G=0 is known a priori? Determinism Lost?
  - Note ∇F=0 is a common terminus to trade-off with all other constraints.
  - Create **Artificial Constraint** H with **Known** starting point ∇H=0
  - $\nabla H=0 \rightarrow \nabla F=0 \rightarrow \nabla G=0$
  - **Determinism is restored!**



Application beyond Matching? – Restoring Determinism

- Algorithm should work on any other function with an analytical model.
- Determinism depends on "known" starting point. (∇H=0, Δk_m=0 or k_m=0)
- What if neither ∇F=0 nor ∇G=0 is known a priori? Determinism Lost?
  - Note ∇F=0 is a common terminus to trade-off with all other constraints.
  - Create **Artificial Constraint** H with **Known** starting point ∇H=0
  - $\nabla H=0 \rightarrow \nabla F=0 \rightarrow \nabla G=0$
  - **Determinism is restored!**

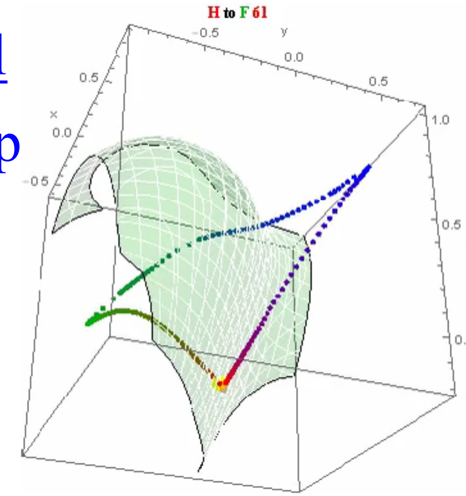


Example of other possible objectives/constraints: (?)

- Beam size at location inside matching section
- Total phase advance
- Weighted mismatch Φ'
- Absolute quad strengths
- Weighted quad strengths (well defined meaning)
- Maximizing mismatch Φ (λ>0)
- Transfer matrix elements
- Special module parameter (e.g., residual dispersion)
- Higher order effects
- Geometric parameters (e.g. Length)
- Quad strings
- Optical functions (e.g., dispersion, chromaticity, ….)

Application beyond Matching? – Restoring Determinism

- Algorithm should work on any other function with an analytical model.
- Determinism depends on "known" starting point. (∇H=0, Δk_m=0 or k_m=0)
- What if neither ∇F=0 nor ∇G=0 is known a priori? Determinism Lost?
  - Note ∇F=0 is a common terminus to trade-off with all
  - Create **Artificial Constraint** H with **Known** starting p
  - ∇H=0 $\rightarrow$ ∇F=0 $\rightarrow$ ∇G=0
  - **Determinism is restored!**



Example of other possible objectives/constraints: (?)

- Beam size at location inside matching section
- Total phase advance
- Weighted mismatch Φ'
- Absolute quad strengths
- Weighted quad strengths (well defined meaning)
- Maximizing mismatch Φ (λ>0)
- Transfer matrix elements
- Special module parameter (e.g., residual dispersion)
- Higher order effects
- Geometric parameters (e.g. Length)
- Quad strings
- Optical functions (e.g., dispersion, chromaticity, ….)

Summary & Future Possibilities

- Possibility to Realize 3 Alternate Views to Matching
  - Distributed instead of Local
  - Deterministic Tradeoff Integration instead of Single objective optimization
  - Offline Computation instead of Online
- Interlinked Concepts – But Not a Monolithic Program to Implement
  - New Matching Engine is enabling component with unique advantages.

New Optimization Algorithm

Rigor
Determinism
Extensive adaptability
Flexibility
Insight

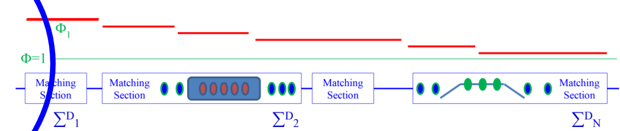
Stand-Alone Matching Engine
→
Can Be Developed Independently

Summary & Future Possibilities

- Possibility to Realize 3 Alternate Views to Matching
  - Distributed instead of Local
  - Deterministic Tradeoff Integration instead of Single objective optimization
  - Offline Computation instead of Online
- Interlinked Concepts – But Not a Monolithic Program to Implement
  - New Matching Engine is enabling component with unique advantages.
  - Distributed Scheme

New Optimization Algorithm

Rigor
Determinism
Extensive adaptability
Flexibility
Insight



Distributed Matching Configuration

Smooth envelope/jitter
Reduced compounded error
Global matching improvement
Recourse to matching error
Low quad strength
Flexibility

- Addresses error accumulation & global mismatch
- Reduced quad strengths
- Countermeasure/Tolerance to matching error
- User flexibility
- All algorithmic advantages

Summary & Future Possibilities

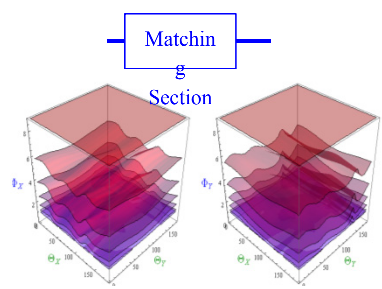
- Possibility to Realize 3 Alternate Views to Matching
  - Distributed instead of Local
  - Deterministic Tradeoff Integration instead of Single objective optimization
  - Offline Computation instead of Online
- Interlinked Concepts – But Not a Monolithic Program to Implement
  - New Matching Engine is enabling component with unique advantages.
  - Distributed Scheme
  - Interpolated Solution

New Optimization Algorithm

Rigor
Determinism
Extensive adaptability
Flexibility
Insight

Solution by Interpolation

Speed
Predictability
Reproducibility
Control flexibility
Insight



- Speed
- Predictability
- Reproducibility
- All algorithmic advantages

Summary & Future Possibilities

- Possibility to Realize 3 Alternate Views to Matching
  - Distributed instead of Local
  - Deterministic Tradeoff Integration instead of Single objective optimization
  - Offline Computation instead of Online
- Interlinked Concepts – But Not a Monolithic Program to Implement
  - New Matching Engine is enabling component with unique advantages.
  - Distributed Scheme
  - Interpolated Solution

New Optimization Algorithm
Rigor
Determinism
Extensive adaptability
Flexibility
Insight

Solution by Interpolation
Speed
Predictability
Reproducibility
Control flexibility
Insight

Distributed Matching Configuration
Smooth envelope/jitter
Reduced compounded error
Global matching improvement
Recourse to matching error
Low quad strength
Flexibility

Best Taken into Account at Design Stage

Summary & Future Possibilities

- ❖ Possibility to Realize 3 Alternate Views to Matching
  - ➢ Distributed instead of Local
  - ➢ Deterministic Tradeoff Integration instead of Single objective optimization
  - ➢ Offline Computation instead of Online
- ❖ Interlinked Concepts – But Not a Monolithic Program to Implement
  - ➢ New Matching Engine is enabling component with unique advantages.
  - ➢ Distributed Scheme
  - ➢ Interpolated Solution
- ❖ Application Beyond Matching
  - ➢ Works on any parameter with well-behaved analytic model
  - ➢ Determinism can be maintained <u>even when start point is not known a priori</u>

  ⇒ (Impose Artificial Constraint)

New Optimization Algorithm
Rigor
Determinism
Extensive adaptability
Flexibility
Insight

Solution by Interpolation
Speed
Predictability
Reproducibility
Control flexibility
Insight

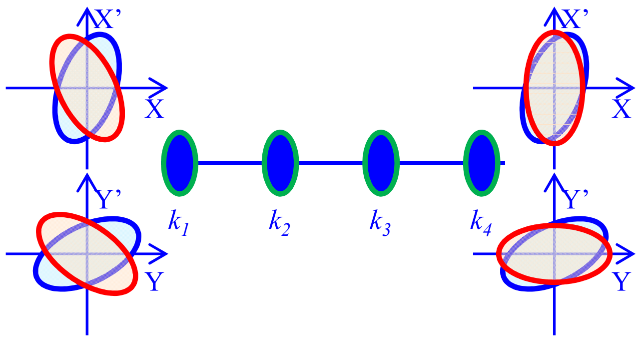
Distributed Matching Configuration
Smooth envelope/jitter
Reduced compounded error
Global matching improvement
Recourse to matching error
Low quad strength
Flexibility

Best Taken into Account at Design Stage

Input / Idea of Application Welcome!

If λ=0, Stop

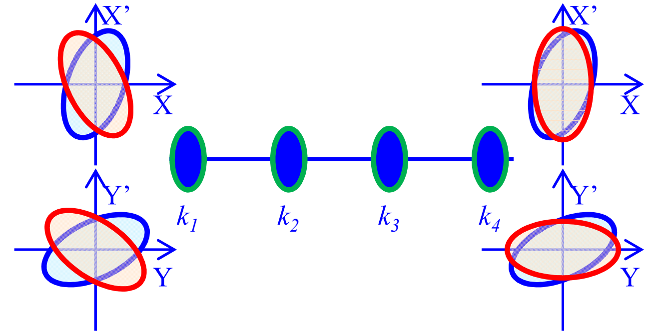
If λ=0, Stop

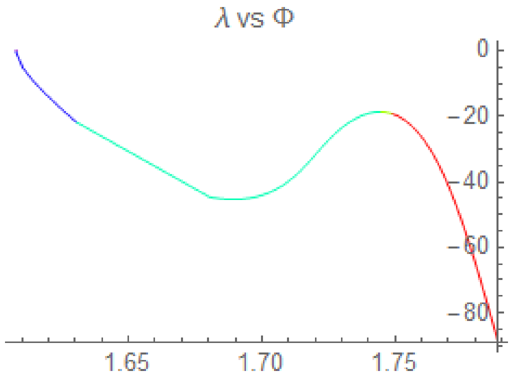


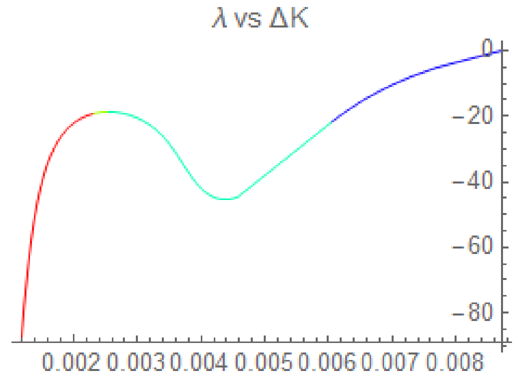
- ❖ 120° FODO; 4 Quad Matching Section;

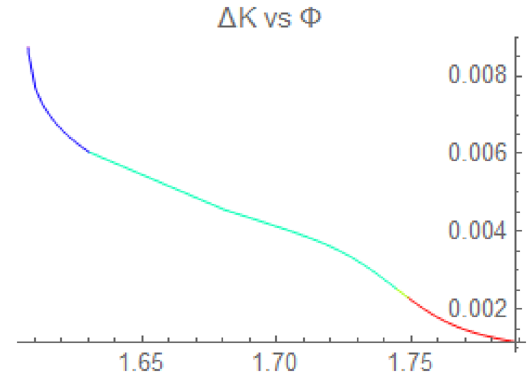
If λ=0, Stop

- 120° FODO; 4 Quad Matching Section;



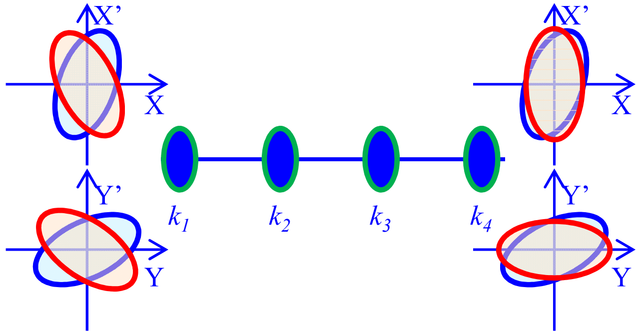


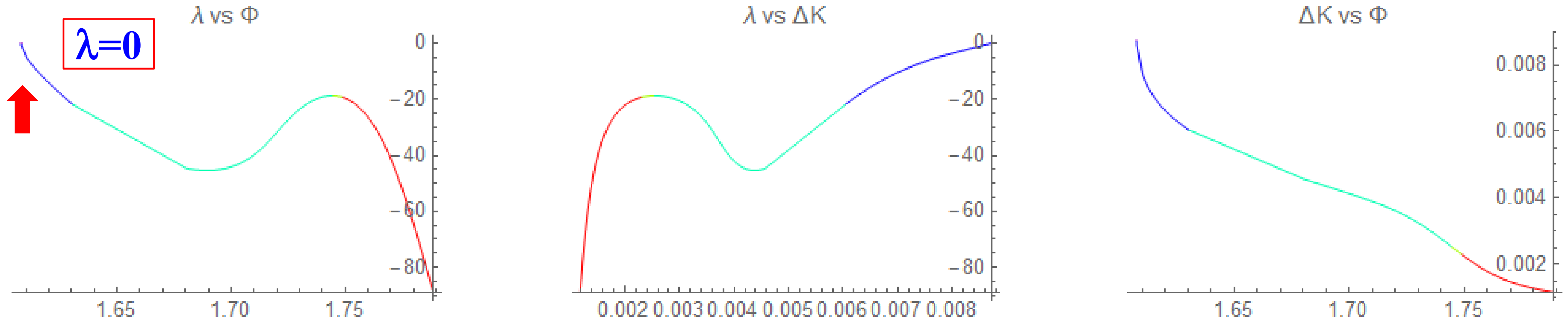




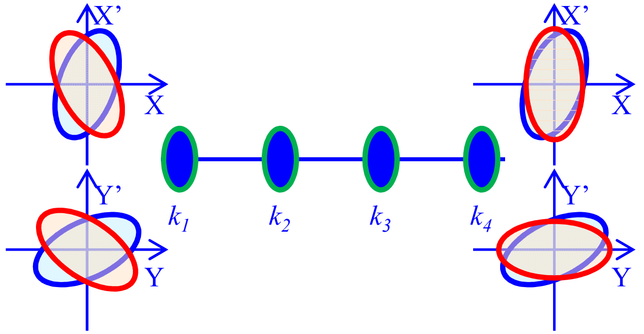
If λ=0, Stop

❖ 120° FODO; 4 Quad Matching Section;

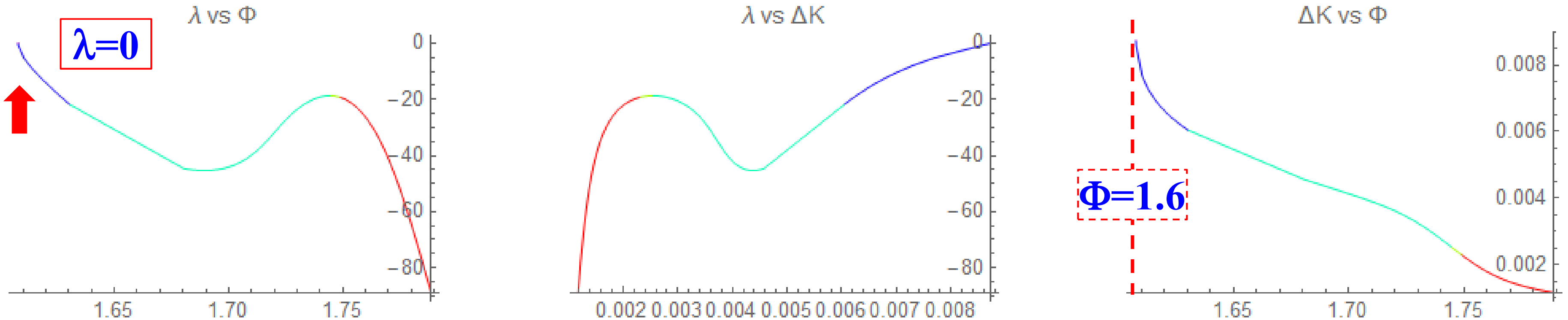




If λ=0, Stop

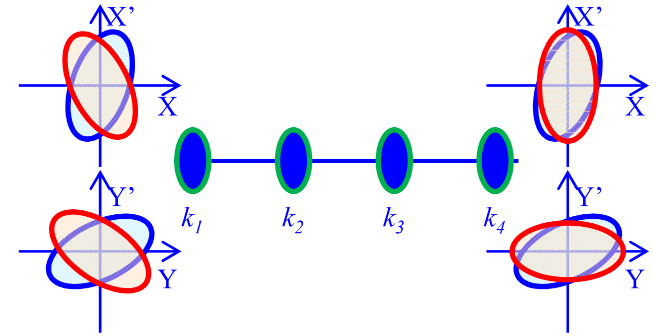


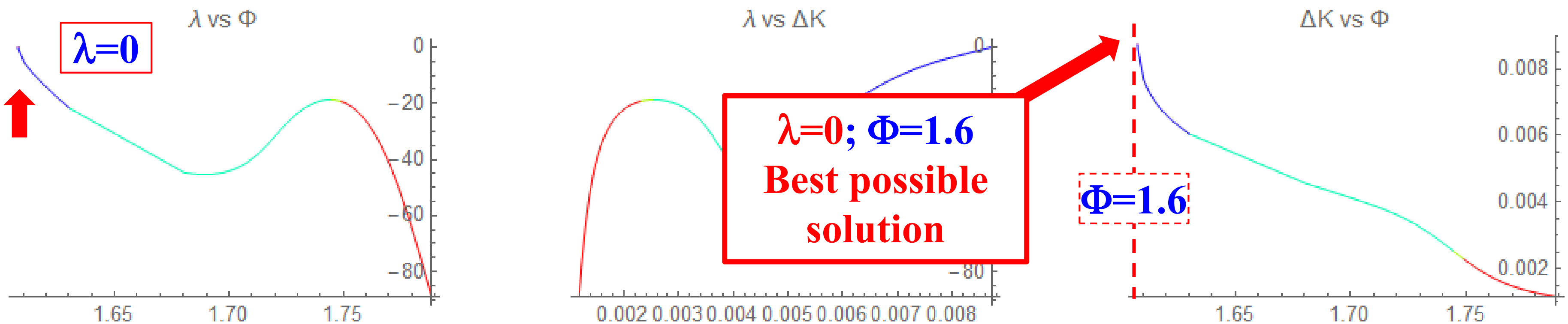
- 120° FODO; 4 Quad Matching Section;



If λ=0, Stop

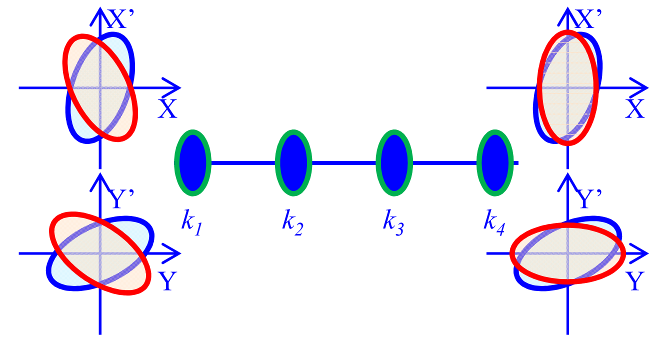
- ❖ 120° FODO; 4 Quad Matching Section;

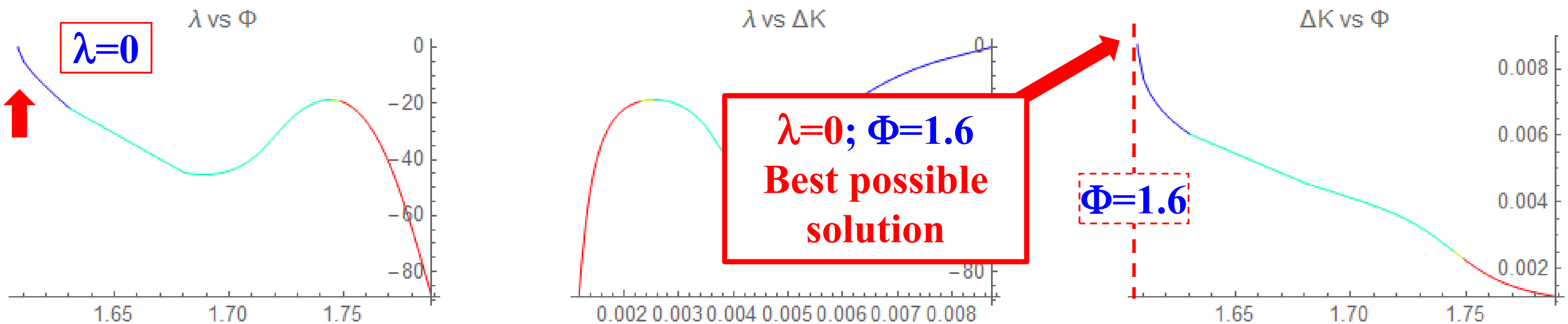




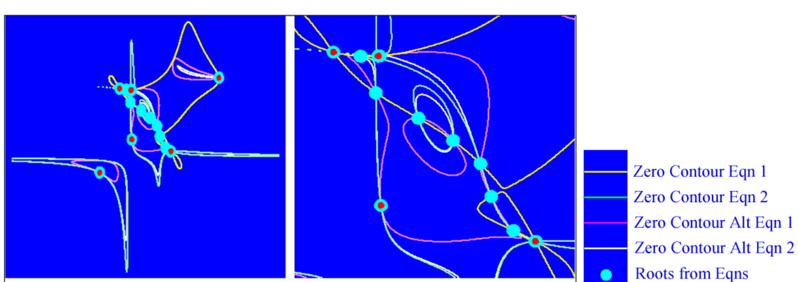
If λ=0, Stop

❖ 120° FODO; 4 Quad Matching Section;

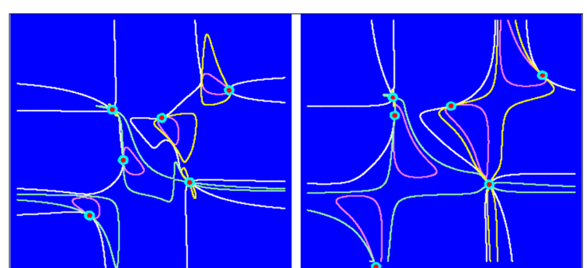




➢ This 4-Quad Mismatched Configuration Does <u>Not</u> Allow 100% Matching (All Roots Are Complex).



Example of Real Roots for a Well-Configured System.

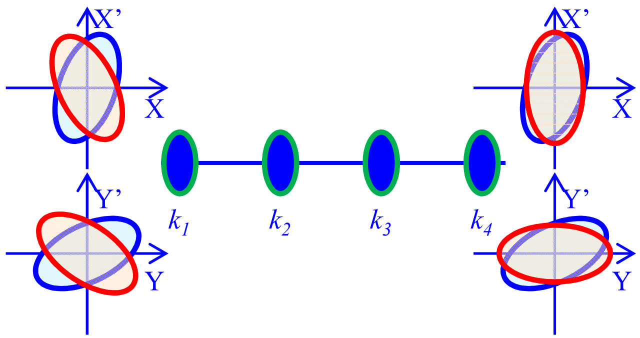


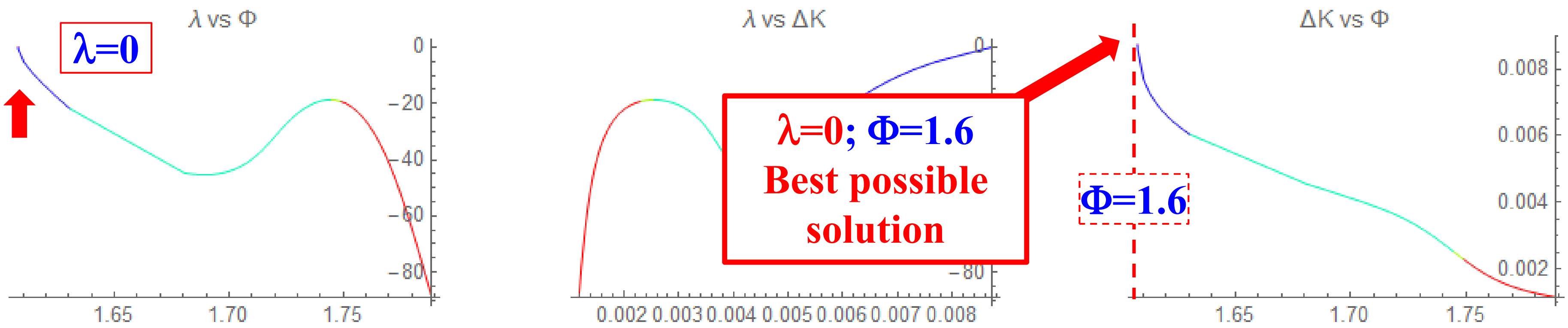
Other Examples of Complete Absence of Real Roots.

4-Quad Matching with
No Real Roots
Y. Chao PAC 2001

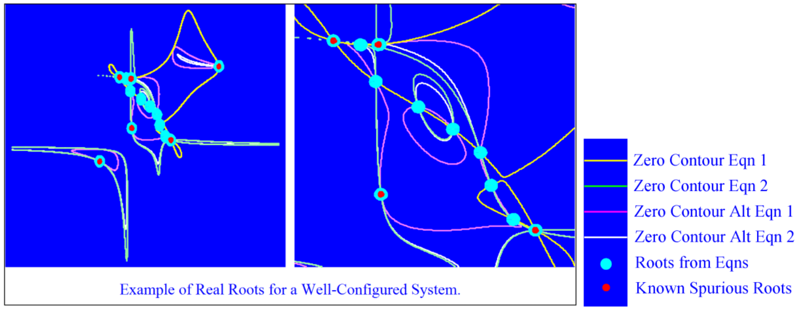
If λ=0, Stop

- ❖ 120° FODO; 4 Quad Matching Section;

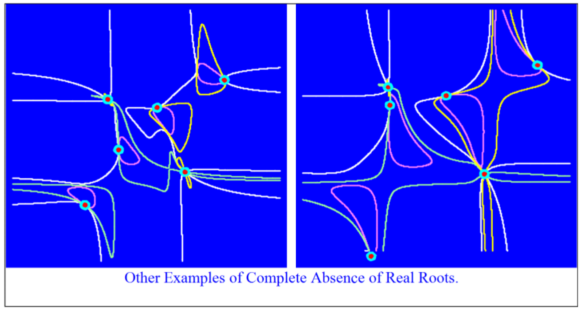




- ➢ This 4-Quad Mismatched Configuration Does Not Allow 100% Matching (All Roots Are Complex).
- ➢ Conventional Algorithm Cannot Give Unequivocal Answer Like This.



Example of Real Roots for a Well-Configured System.


Other Examples of Complete Absence of Real Roots.

4-Quad Matching with No Real Roots
Y. Chao PAC 2001