Argonne NATIONAL LABORATORY

... for a brighter future

Optimization Algorithms for Accelerator Physics Problems

10th International Computational Accelerator Physics Conference
ICAP-09, August 31- September 4, 2009
San Francisco, California, USA

B. Mustapha and P. Ostroumov

Physics Division

Argonne National Laboratory

U.S. Department of Energy

UChicago Argonne LLC

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

Outline

- Optimizations in Accelerator Physics are Used / Needed for …
- Elements / Ingredients of an Optimization Problem
- Local versus Global Optimizations
- Standard versus Evolutionary Algorithms
- Parallel Optimizations
- Applications in Beam Dynamics Optimization
- Potential Application: The Model Driven Accelerator
- Summary

Optimizations in Accelerator Physics are Used / Needed for …

- **Accelerator Design: Optimizations are heavily used in the design phase**
  - Element design optimization: RF cavities, magnets, …
  - Lattice optimization: Sequence of elements, drift spaces, …
  - Beam dynamics optimization: matching, beam quality, …

- **Commissioning: More effort is being devoted to support commissioning**
  - Help better understand the machine's behavior → Deliver the 1st beam
  - Fits to reproduce the data using a model
  - Improve the predictability of the model to hopefully use for operations

- **Operations: Often simplified models (1D, single particle) are used for speed**
  - Fits to support machine tuning / retuning
  - Detailed 3D codes are used off-line

Elements / Ingredients of an Optimization Problem

- **Objective (s): Important quantities/qualities, characteristics of the problem**

  - Functions you would like to optimize

- **Parameters: Variables affecting the outcome of the problem (objectives).**

  - If too many parameters, choose the parameters to which the problem is more sensitive.

- **Parameters constraints and correlations: Define the parameter space**

  - The simplest: Independent parameters with lower and upper bounds

  - If correlated: Try reduce to an independent set of parameters

- **Optimization algorithm: Local, Global, Standard, Evolutionary …**

➢ **Proper definition of the problem is an important first step …**

Local versus Global Optimizations

- **It is not very hard to find a local minimum of an objective function,**
  What is hard is to prove that it is the best minimum,
  It is even harder to prove that a minimum is a global one.

- **A Local Optimizer**
  - Starts with a first guess
  - Finds a direction that minimizes the objective and moves one step
  - The procedure is repeated iteratively until no progress could be made
  → Fast

- **A Global Optimizer**
  - Should explore the full parameter space
  - Eventually finds all local minima before finding a global one
  - Prove that the minimum found is a global one
  → Slower

Local versus Global Optimizations (2)

- **Luckily not all problems / applications require global optimizations**

- **A global optimizer is more appropriate for design optimization to map the whole parameter space.**
  - **- You don't want to miss the best set of design parameters**
  - **- Find all feasible solutions and make compromises if needed**

- **A local optimizer with a shorter path to solution is more adequate for accelerator operations.**
  - **- Start from a good starting point**
  - **- Find the next best operating point by retuning few elements**

➢ **Time to solution is a very important parameter: A good optimizer should also optimize the path to the best solution …**

Standard versus Evolutionary Algorithms

- **Standard Algorithms**
  - Most common, widely used
  - A single objective function to optimize (could be a weighted sum)
  - Usually require objective function derivatives w.r.t. the parameters
  - A single trial solution is evaluated at every iteration (local)

- **Evolutionary Algorithms**
  - Based on the theory of evolution and natural selection, only the best survive
  - Multiple objective functions could be included in the optimization
  - Do Not require objective function derivatives
  - Multiple trial solutions are evaluated at every iteration (global)

Examples of Standard Algorithms

- **Simplex method (s): Does not require derivatives**
  - Build a simplex: First guess + base of feasible solutions in parameter space
  - Iteratively: Replace the worst solution by a better one using the base
  - Stop: No progress or cut-off error

- **Gradient descent method (s): Uses first derivatives (Gradient)**
  - Search direction at iteration i: $p_i = -B_i^{-1}\nabla f_i$
  - f is the objective and B is a symmetric non singular matrix
  - $B = I$: Identity matrix in the steepest descent method

- **Newton method (s): Uses first and second derivatives (Gradient & Hessian)**
  - $B = H$: Hessian, matrix of second derivatives

- **Quasi-Newton method (s): Uses first derivatives but updates second derivatives after every iteration.**
  - B : Approximation to the Hessian matrix

Example of Evolutionary Algorithm: Genetic Optimizer

1) Starts with a set of solutions randomly generated inside the parameter space

2) The solutions are evaluated and ranked based on the objectives and constraints of the problem to select a subset of best solutions.

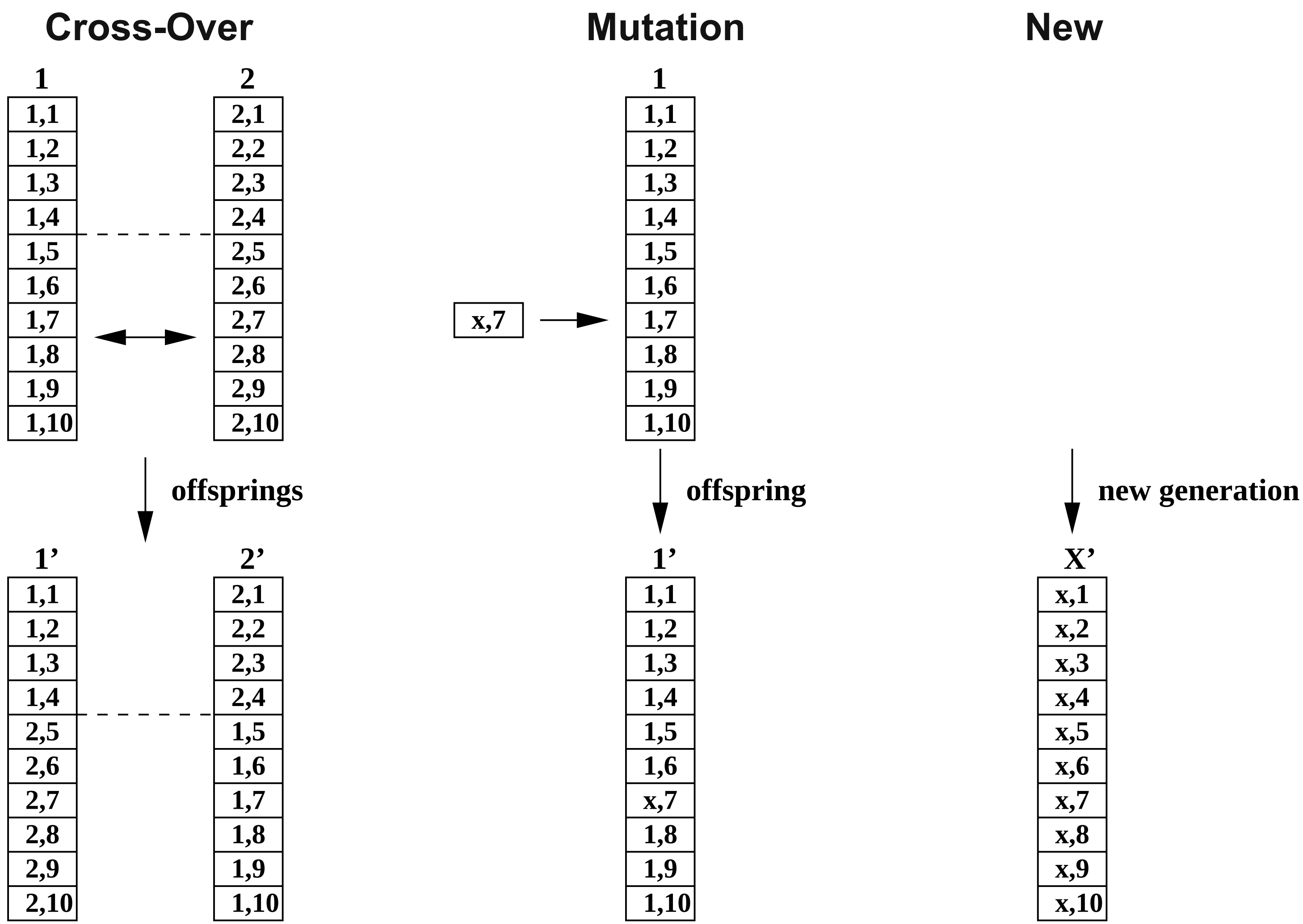
3) The selected solutions are used to generate the next population by crossover, mutation or other using predefined rates (adjustable).

4) Start over from (2)

5) Stop when no progress could be made (stable set of best solutions).

✓ For a given solution, the array of parameter values plays the role of a gene

Genetic Optimization: Generating the next population



✓ **The probability or rate of every channel could be adjusted …**

Parallel Standard Algorithms

- **Standard algorithms are serial in nature:** The direction of the next iteration is decided based on the outcome of the current one.
- **Single solution evaluated per iteration:** Parallelizable at the level of function and derivatives evaluation.
- **Example 1:** Least square minimization:
  Parallelize the sum for large N

$$\sum_{i=1}^{N} \frac{\Delta X_i^2}{\varepsilon_X^2}$$

- **Example 2:** Optimization with a multi-particle tracking code → Parallel particle tracking, Poisson solver and statistics (Large number of particles).
- **Example 3:** A global optimizer may be parallelized by sub-dividing the parameter space and assigning the different sub-spaces to different processors.

Parallel Evolutionary Algorithms

- **Multiple solutions are evaluated independently at every iteration:** **Well suited for parallel processing with minimal communication.**
- **It is however Not easy to parallelize the ranking, selection and offsprings generation:** **Usually assigned to the master process.**
- **More appropriate for optimization using multi-particle codes with realistic 3D external and space charge fields.**
- **No parallel particle tracking is required unless a very large number of particles is needed for the optimization problem.**
- **Any particle tracking code with SC could be used:** **A higher level parallel layer is often used to manage the generation, ranking and selection of trial solutions and calls the code when needed.**
- **In our beam dynamics code TRACK, it was built-in.**

Applications: Beam Dynamics Optimization

Multiple Charge State Ion Beam: Longitudinal tuning before a stripper

- **Purpose:** Tune a linac section to minimize the longitudinal emittance of a multiple charge state beam right before stripping.
- **Method:** Match the longitudinal beam centers and Twiss parameters of the different charge state beams:

$$W_{q0} \rightarrow W_0;\ \Delta W_{qi} \rightarrow 0;\ \Delta\phi_{qi} \rightarrow 0;\ \alpha_{qi} \rightarrow 0;\ \beta_{qi} \rightarrow \min$$

- **Fit Function:**

$$F = \frac{(W_{q0} - W_0)^2}{\varepsilon_w^2} + \sum_{qi} \frac{\Delta W_{qi}^{\ 2}}{\varepsilon_{\Delta W}^2} + \sum_{qi} \frac{\Delta\phi_{qi}^{\ 2}}{\varepsilon_{\Delta\phi}^2} + \sum_{qi} \frac{\alpha_{qi}^{\ 2}}{\varepsilon_{\alpha}^2} + \sum_{qi} \beta_{qi}$$

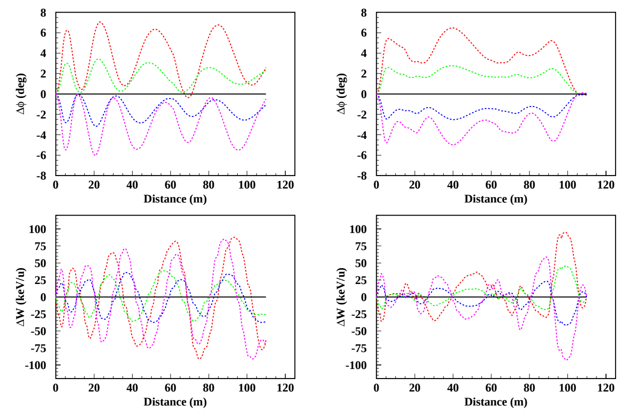
where W_0 is the desired beam energy and ε_W is the corresponding error.

$\varepsilon_{\Delta W}, \varepsilon_{\Delta\phi}, \varepsilon_{\alpha}$ are the allowed errors on the relative energy, phase and α shifts of the individual charge state beams from the central beam.

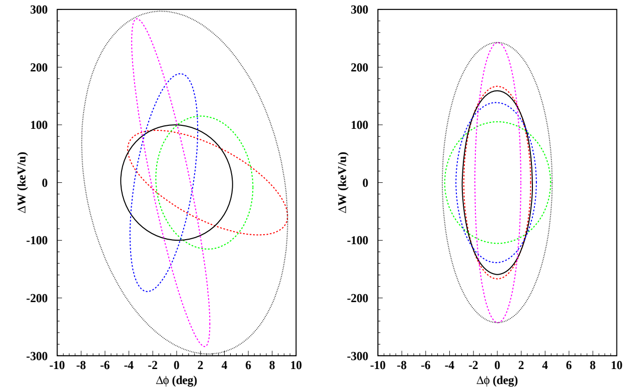
- **Fit Parameters:** RF cavities field amplitudes and phases.

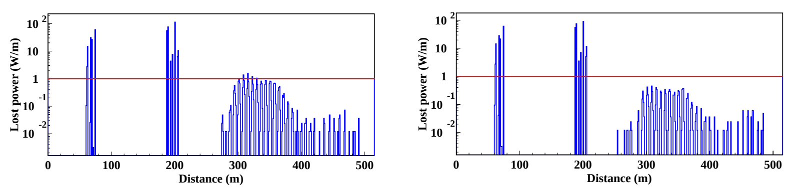
➢ **Measuring the energy and phase of individual charge states, we should be able to match their beam centers, …**

✓ **B. Mustapha and P. Ostroumov, Phys. Rev. ST Accel. Beams 8, 090101 (2005)**



Black: Ref. 74+ of U-238
Colors: 72+,73+,75+ and 76+





Reduced beam loss in the high-energy section

Realistic Corrective Steering: Front-end of the FNAL-PD

Algorithm

Consider an accelerator section with Nm monitors and Nc correctors

↓

Determine the response function of monitors to correctors
In the matrix form: M = F(C) = A*C+B
M: array of monitors readings
C: array of correctors strengths
A: response function matrix
B: monitors readings for C=0

↓

Solve the equation A*C + B = 0 for C
To have the beam centered on all monitors M=0

↓

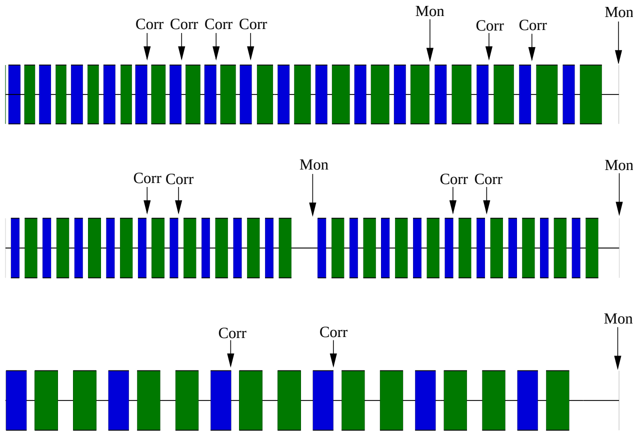
Apply the values of C to correct the beam

➢ In TRACK instead of solving the matrix equation A*C+B = 0 for the correctors strength C, we perform a least square minimization of the equivalent function:

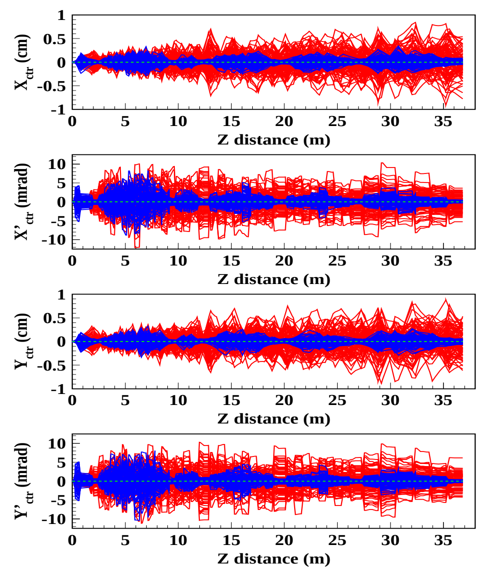
$$f(C_{i_c}, i_c = 1, N_c) = \sum_{i_m=1}^{N_m} \frac{\left(\sum_{i_c=1}^{N_c} A_{i_c i_m} * C_{i_c} + B_{i_m}\right)^2}{\sigma_{i_m}^{\;2}};\ for \left|C_{i_c}\right| \le C_{\max}$$

➢ In this way, we can include the monitors precision σim and the maximum correctors strength Cmax in the solution. Monitors with different precisions will have different weights.

Locations of monitors and correctors

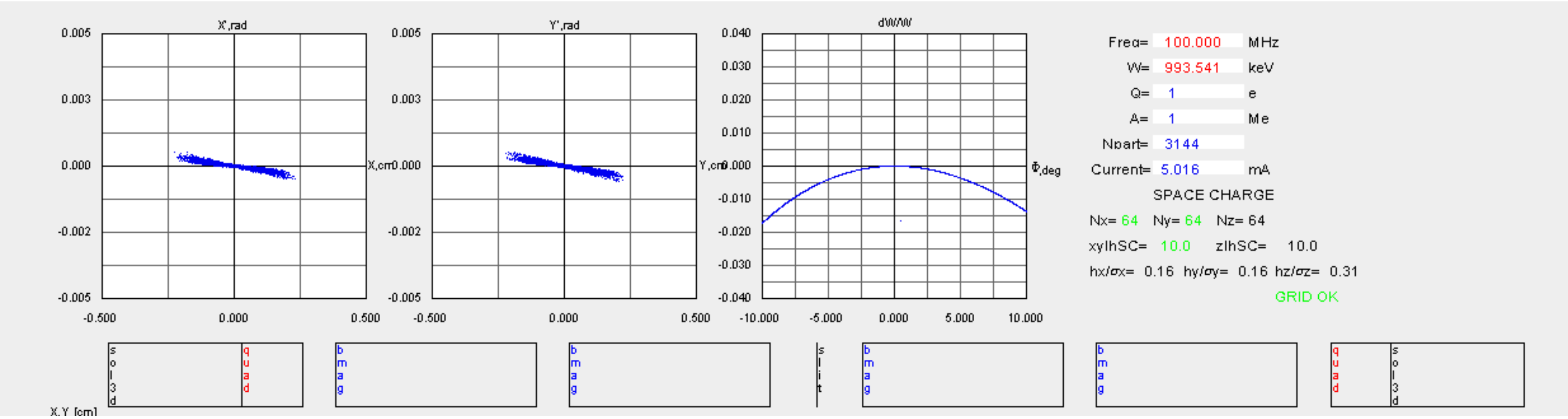


Beam centeroids before and after corrections



Genetic Optimization: A Chicane in ultra-low emittance e-Injector

Minimize the transverse emittance growth by fitting the quads and solenoids strengths.



Manually optimized case : ε(x, 80%) = 0.090 μm, ε(y, 80%) = 0.092 μm

Genetic optimization result: ε(x, 80%) = 0.078 μm, ε(y, 80%) = 0.081 μm → ~ 10% less

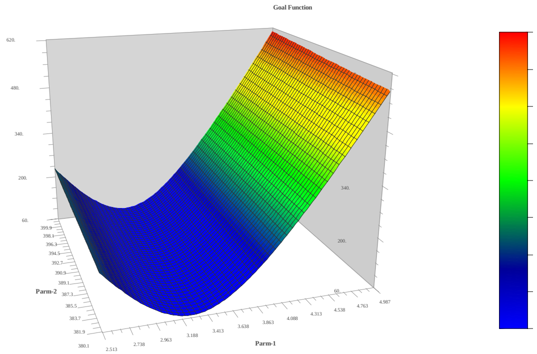
A Convergence Test

A 2D problem

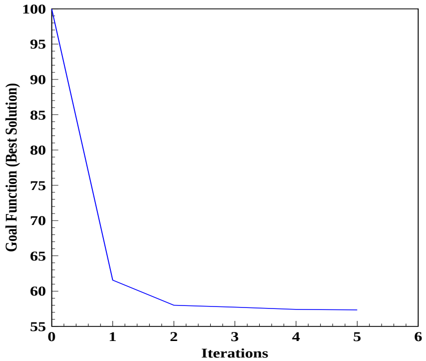
1 Objective

2 Parameters

2D Map: 10000 solutions



G-Optimizer: 600 solutions



Potential Application: Model Driven Accelerator

Model Driven Accelerator: Concept & Motivations

- **The Concept:** Use a computer model to fully support real-time accelerator operations.

- **Present Situation:** No accelerator in the world could fully rely on a computer model for its operations.

- **Possible Reasons:** Discontinuity between the design and operations phases
  - Design and simulations assume almost perfect conditions.
  - Elements specs are usually different from their original design.
  - Not enough diagnostics to characterize the machine.

- **Consequences:** Delay in commissioning and low machine availability.
  - Simulations cannot reproduce the measured data.
  - A lot of work to deliver the first beam during commissioning.
  - A lot of time spent on beam tuning/retuning during operations.

Model Driven Accelerator: Concept & Motivations

- **For Example:** RIA / FRIB Cannot afford "manual" operations …
  - Primary beams: p to U, from 200 MeV/u to 600 MeV/u
  - Secondary beams: all over the map …

➢ **Need a realistic computer model for the machine to support commissioning and operations**

- **The Benefits:**
  - Fast tuning for the desired beam conditions.
  - Fast retuning to restore the beam after a failure.
  - Increase the availability of the machine.
  - Reduce the operating budget.

- **The Means:**
  - A realistic 3D model of the actual machine.
  - Fast turn-around optimizations to support decision making.

Realization of the Model Driven Accelerator: What we need ?

- **Need a realistic 3D beam dynamics code with the appropriate set of optimization tools and large scale parallel computing capability.**

- **Why a beam dynamics code ?**
  - More realistic: 3D fields including fringe fields and SC calculations
  - More detailed: Beam halo, beam loss, …
  - Produce detector-like data: Profiles, distributions, …

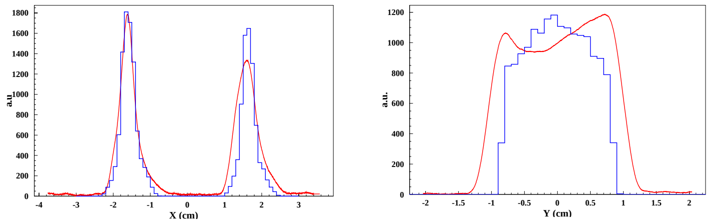
- **Why more optimization tools ?**
  - Optimization tools are needed not only in the design phase but also to tailor the model to the actual machine to be used for real-time operations.

- **Why large scale computing ?**
  - Optimizations of large number of parameters with a large number of particles for large number of iterations require large scale computing.

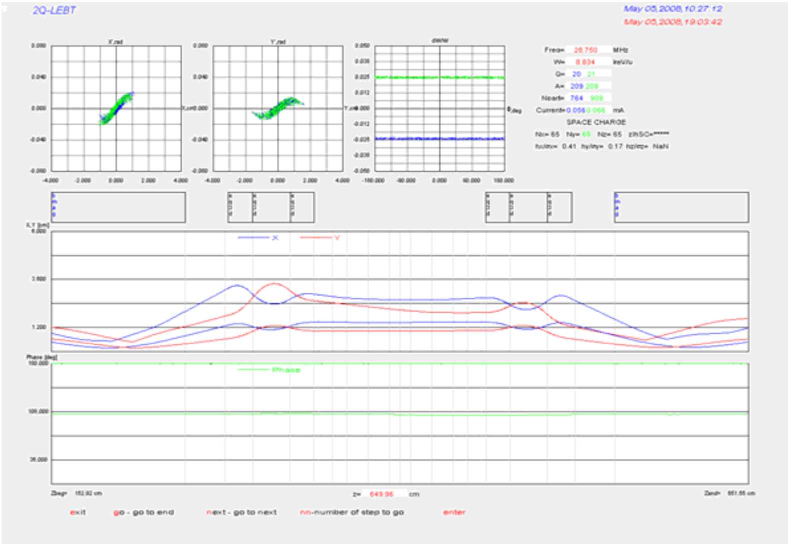
- **The beam dynamics code TRACK is being developed at Argonne to meet these requirements.**

Small Scale Realization of MDA: Operations of a Multi-Q Injector

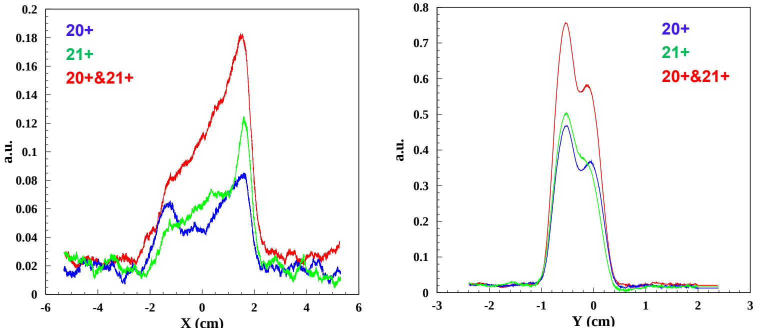
TRACK fit of measured profiles to extract the initial beam parameters at the source.



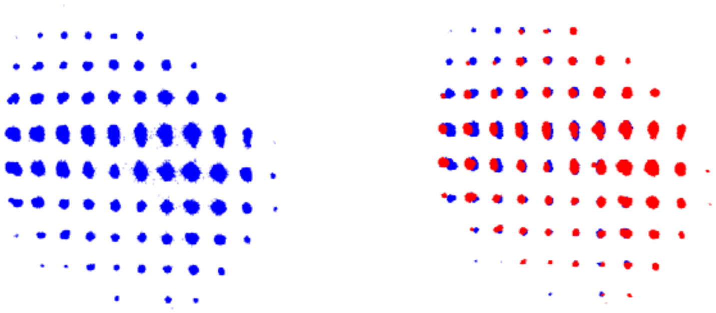
TRACK fit to find the quads setting to recombine the two charge state Bi-209 beams at the end of the LEBT.



Measured beam profiles at the end of LEBT: left: horizontal, right: vertical.



Pepper-Pot images: Bi-209 beams left: 20+&21+ right: 20+: blue, 21+:red.



➢ **Such a perfect recombination was not possible without a realistic simulation.**

✓**P. Ostroumov, S. Kondrashev, B. Mustapha, R. Scott and N. Vinogradov, Phys. Rev. ST-AB 12, 010101 (2009)**

Further Developments towards MDA

- To use a realistic 3D model for real-time machine operations we should be able to perform large scale optimizations on large number of processors.
- The parallel version of TRACK is now ready, parallel optimization tools are being developed: Different algorithms are being implemented.
- Develop more tools for the commissioning phase to tailor the computer model to the actual machine by fitting the measured data.
- Develop interfaces between the beam diagnostic devices and the beam dynamics code → Calibrate and analyze the data to input to the code.
- Numerical experiments may be used to test the tools before implementation into the real machine → Produce detector-like data from the code.
- Application to existing facilities …

Summary

- Optimization tools and methods are needed in every phase of an accelerator project, namely the design, commissioning and operations.
- No single algorithm could satisfy all the optimization needs.
- Different algorithms are being used: local, global, standard, evolutionary, ..
- We briefly reviewed and compared different classes of algorithms and presented few applications in beam dynamics optimization.
- The ultimate goal of realizing the concept of “Model Driven Accelerator” will require a realistic 3D beam dynamics code with the appropriate set of optimization tools and large scale parallel computing capabilities.
- For a new machine we should take advantage of the commissioning phase to bridge the gap between the original design and the actual machine by tailoring the computer model to the machine.
- A significant development effort is still needed …