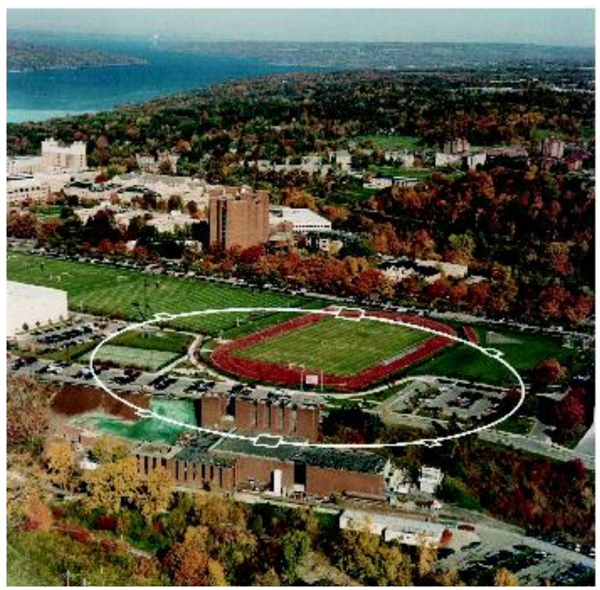
Accelerator & X-ray Modeling Using the Bmad Software Library

David Sagan
Cornell Laboratory for Accelerator-Based Sciences and Education

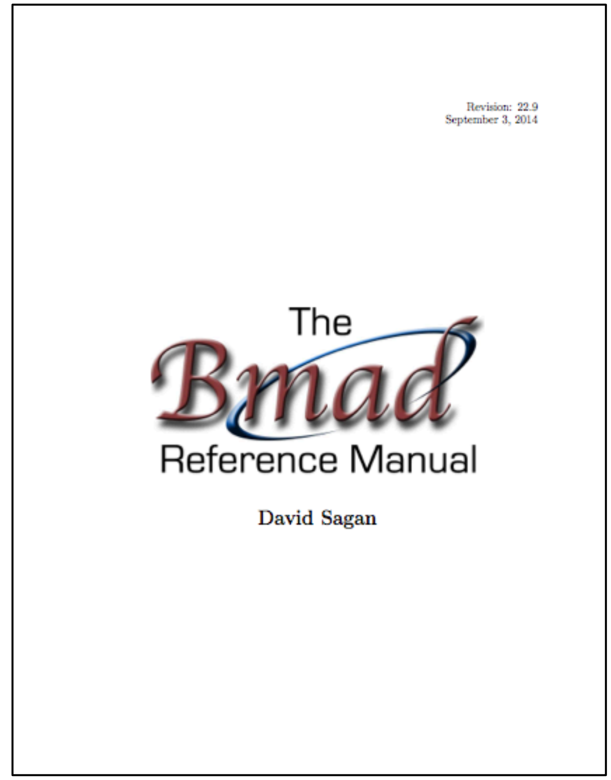


Talk Outline

- Overview & history
- Tao program
- Useful features
- Bmad ecosystem of programs
- Future plans
- Conclusion

Overview

- Bmad is a library toolkit. Not a program
- Written in Fortran.
- Object oriented from the ground up.
- Has structure translation code for interfacing with C++.
- Lattice files use a MAD like syntax.
- Documented.
- Open Source:
  http://www.lepp.cornell.edu/~dcs/bmad/


Revision: 22.9
September 3, 2014

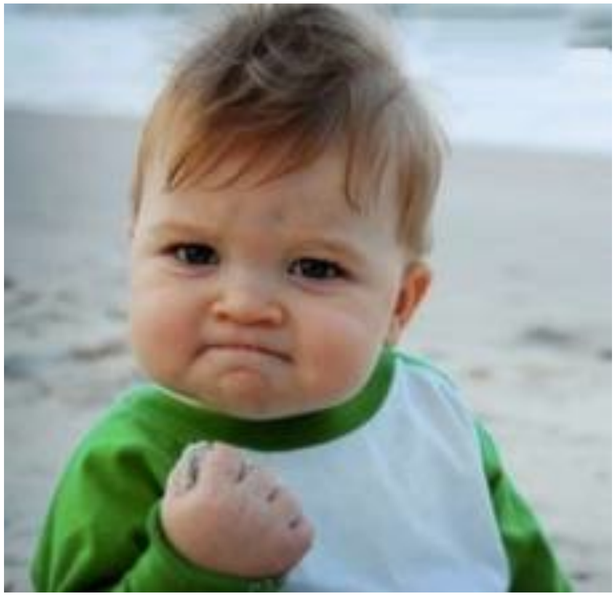
The Bmad Reference Manual

David Sagan

In the Beginning…

Brief History:

- Born at Cornell in mid 1990's
- Started life as modest project: Just wanted to calculate Twiss functions and closed orbit.
- Initially Bmad used a subset of the MAD lattice syntax. Hence the name: "Baby MAD" or "Bmad" for short.



Over the years Bmad had evolved…

And Baby Grows Up...

Currently:

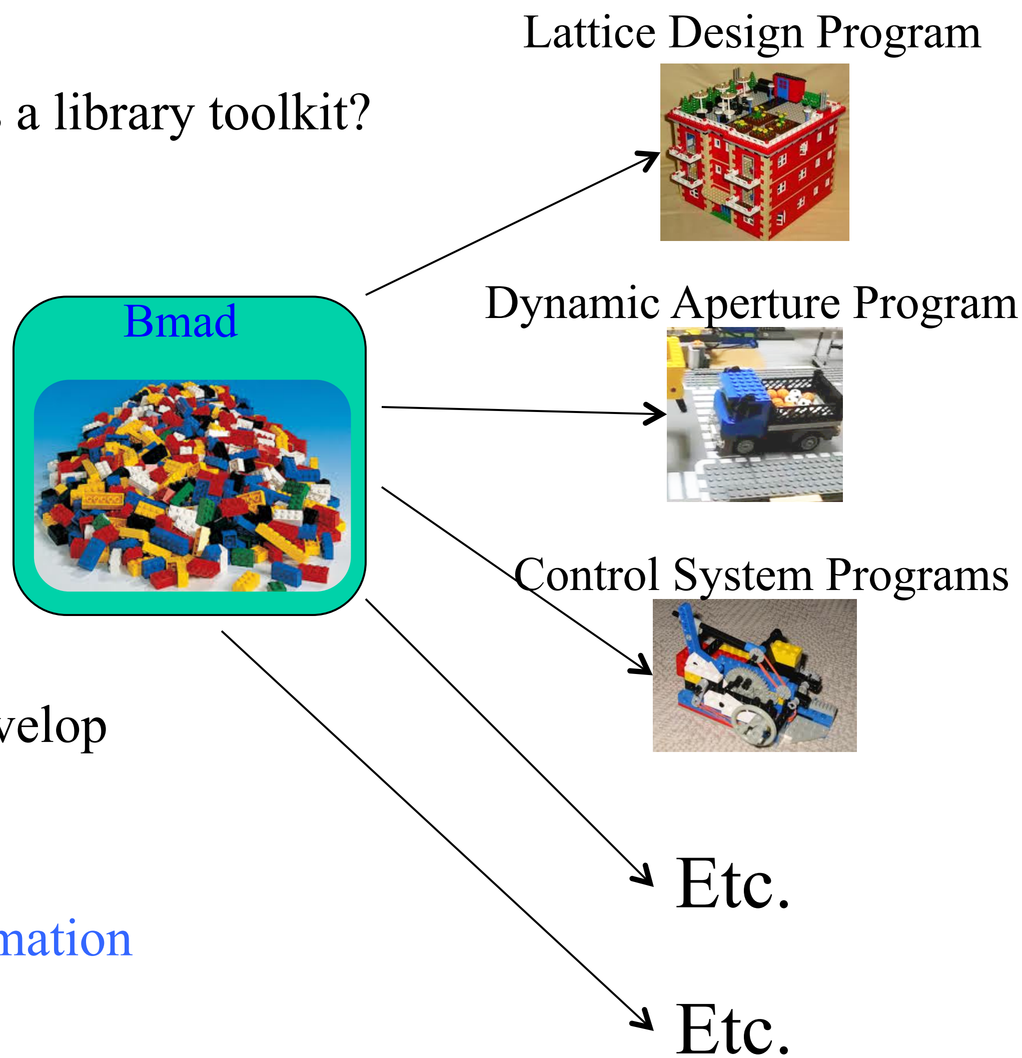
- ~100,000 lines of code
- ~1,000 routines

And it can do much more:

- X-ray simulations
- Spin tracking
- Tracking with coherent synchrotron radiation
- Wakefields and HOMs
- Beam breakup simulations in ERLs
- Intra-beam scattering (IBS) simulations
- Touschek lifetime
- Frequency map analysis
- Dark current tracking
- Etc., etc.

Bmad Philosophy

Question: Why is Bmad constructed as a library toolkit?
Answer: Flexibility!



Advantages of a toolkit:

- Cuts down on the time needed to develop programs.
- Cuts down on programming errors.
- Standardizes sharing of lattice information between programs.

Tao: Tool for Accelerator Optics

Problem: Bmad is not a program so it cannot be used “out of the box.” for simple calculations.

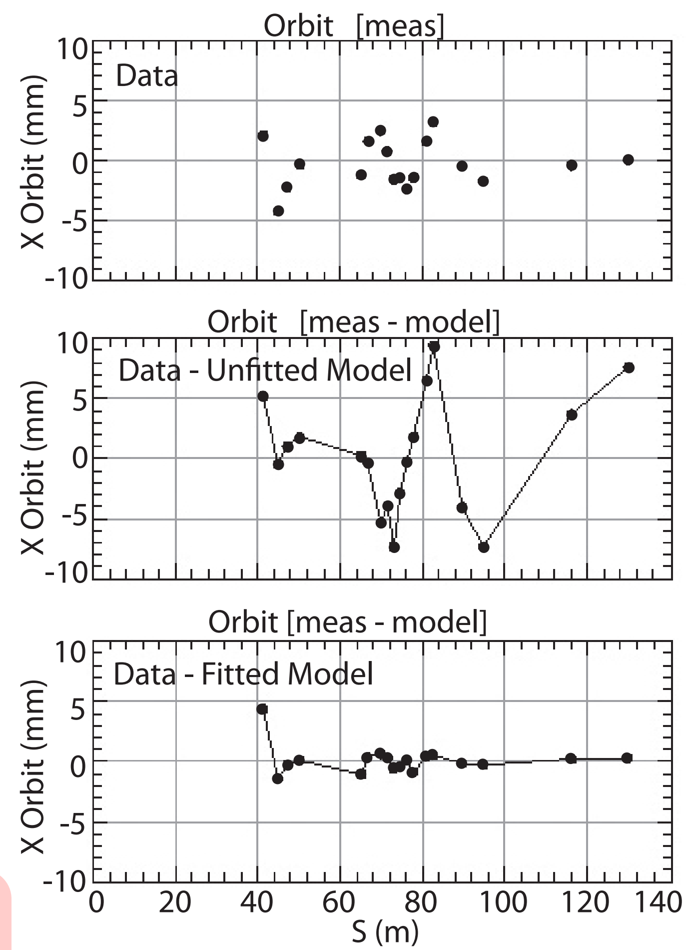
Solution: Develop Tao - a general purpose simulation & design program with

- Nonlinear optimization.
- Twiss and orbit calculations
- Etc.

Additionally: Tao’s object oriented coding makes it relatively easy to extend it.

- For example: Can add custom commands to interface Tao with a control system.

Tao with Bmad gives the flexibility of a library with the convenience of a program.



JLab FEL Modeling

Bmad Features

Bmad has a number of features that over the years have proven useful...

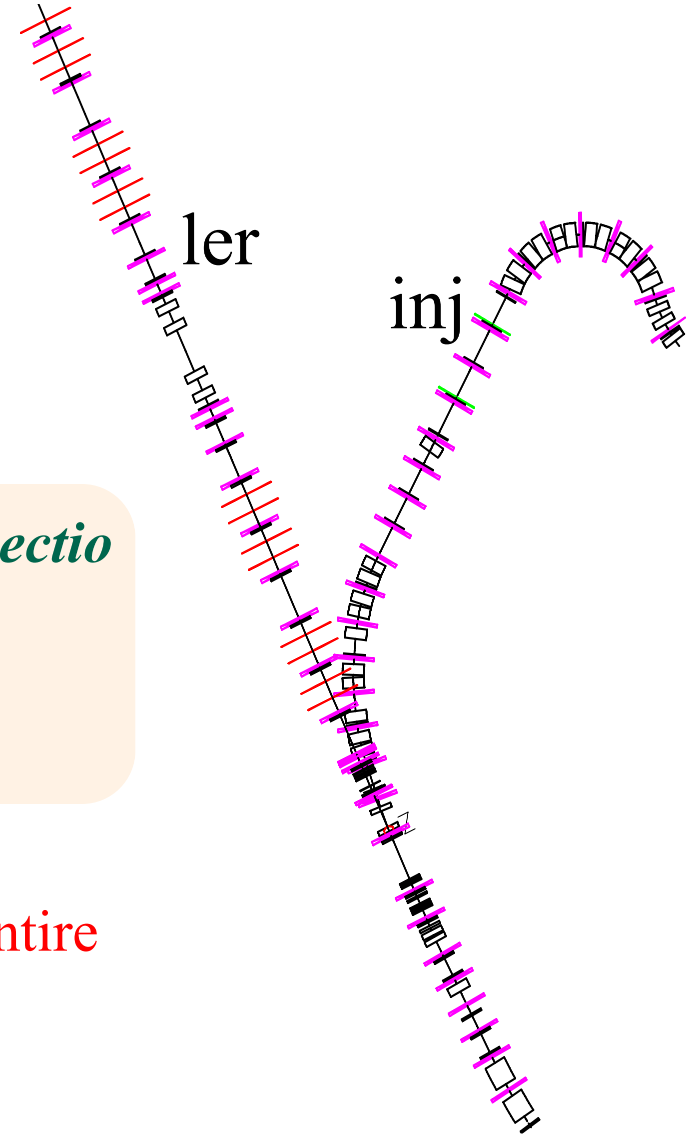
Forking

Bmad can connect different beam lines together. Can join LINACs to injection lines to storage rings to X-ray beam lines, etc.

Example: SuperKEKB LER injection

```
ler_fork: fork, to_line = ler, to_element = injectio
inj: line = (..., ler_fork)          ! injection line
ler: line = (..., injectio, ...)     ! LER ring
use, inj
```

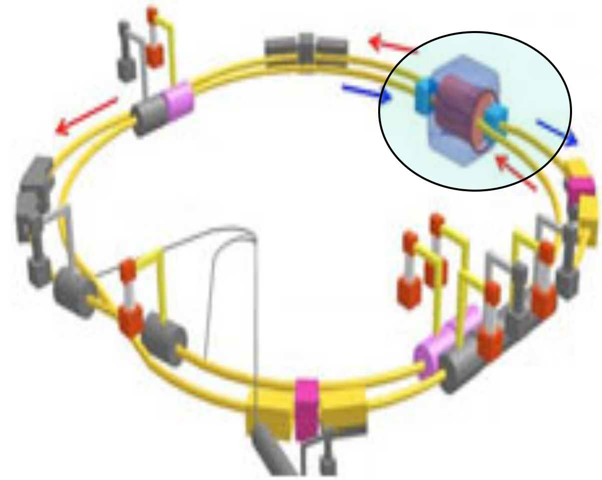
➔ One lattice can hold the description of the entire accelerator complex.



Multipass

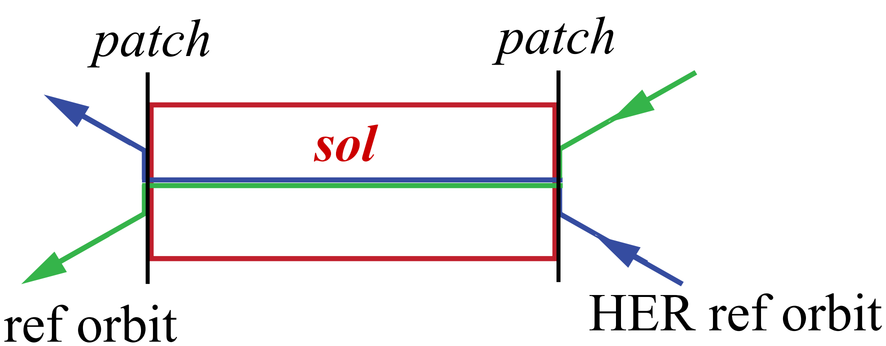
Bmad can simulate beam lines having common elements
Example: SuperKEKB IR region:

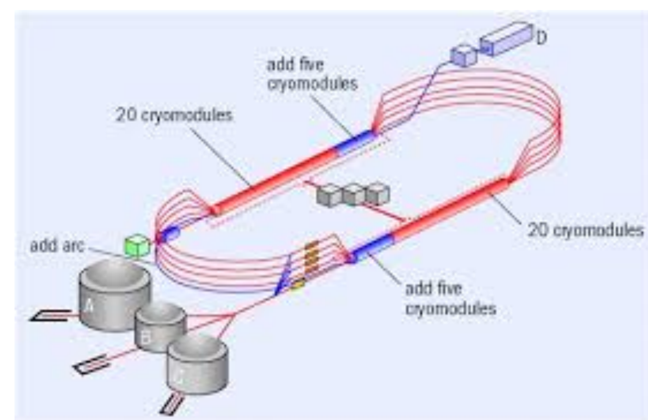
sol: line[**multipass**] = (esle4000, ..., esre4000)
pt1_l: **patch**, ...;   pt2_l: **patch**, ...
pt1_h: **patch**, ...; pt2_h: **patch**, ...
mm: marker
fid: **fiducial**, origin_ele = ***mm***
ler: line = (pt2_l, ***mm***, ***sol,*** pt1_l, ler_arc)
her: line = (pt2_h, --***sol***, ***fid,*** pt1_h, her_arc)
use, ler, her



SuperKEKB

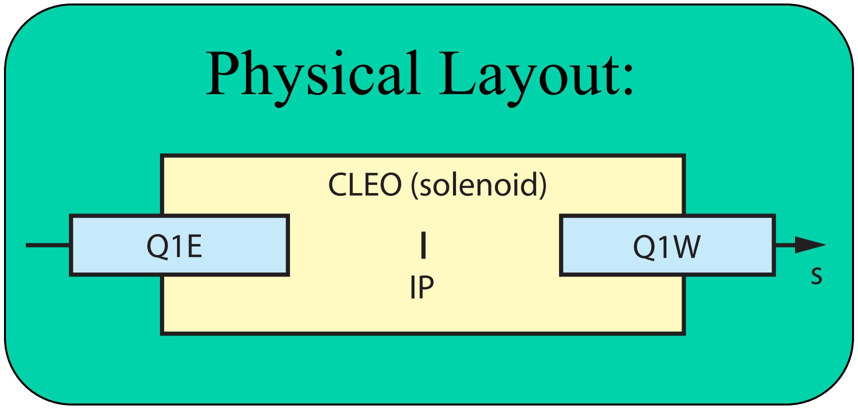
Changes to elements in sol get reflected in both HER and LER lines.

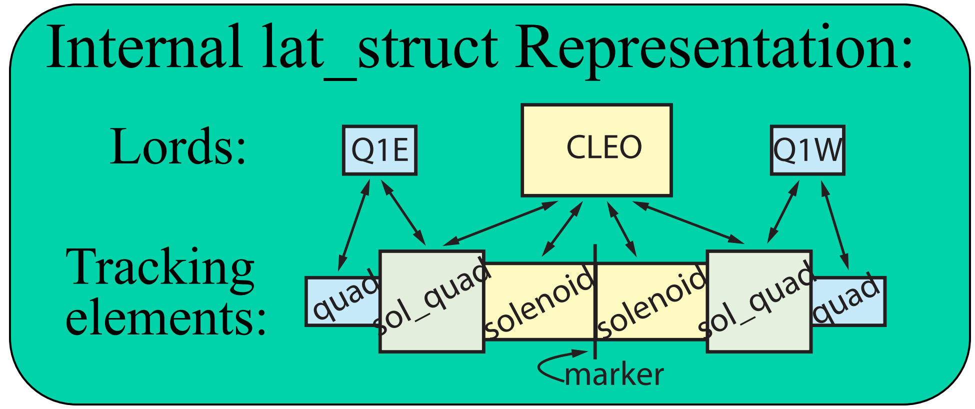




CEBAF

Superposition





Superposition allows element overlap.
In the lattice file:

```
cesr: line = (... q1e, dft, ip, dft, q1w ...)
cleo: solenoid, l = 3.5, superimpose, ref = ip
```

And Bmad does the bookkeeping…

Simplifies life for both user and programmer:

- Simplifies lattice file construction.
- Simplifies varying element attributes in a program.

Tracking Method Selection

Can set how each element is tracked on an element-by-element basis:

• **bmad_standard**	Fast, nonsymplectic
• **symp_lie_ptc**	Symplectic tracking using PTC
• **taylor**	Taylor map (generated via PTC)
• **linear**	Linear tracking
• **custom**	Tracking with custom code
• **runge_kutta**	Track through fields.
• **etc.**	

Example:

inflector: em_field, tracking_method = runge_kutta, field_calc = custom, ...

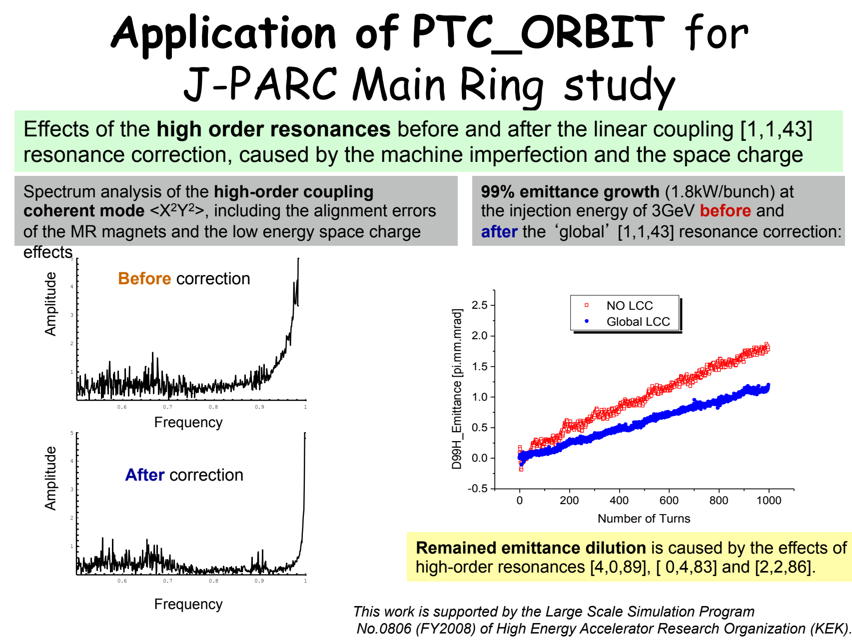
Advantages:

- Enables simulation of unique element types
- Can compare different tracking methods

Etienne Forest's FPP/PTC

Etienne Forest's FPP/PTC simulation toolkit:

- Can construct Taylor series maps to arbitrary order via symplectic Lee integration through elements.
- Can include spin tracking.
- Can do such things as normal form analysis to extract resonance strengths.
- Is used in MAD-X & PTC_ORBIT



Alexander Molodozhentsev

PTC_ORBIT combined code

KEK & SNS common activity

Integration with Bmad:

- Interface routines between Bmad and PTC allow tracking of individual elements
- PTC lattices can be constructed for analysis
- In cooperation with Etienne, PTC now can handle KEKB lattices

Controller Elements

In a Bmad lattice file, can define controller elements [called overlay and group] that control the parameters of other elements.

Example: Power supply powering the quadrupoles in a 2.5π cell in SuperKEKB LER

```
call, file = sler_1689.bmad
cq1 = -0.432108; cq2 = 0.390903
ps1: overlay = {qd1p##1[k1]:cq1,
        qf2p##1[k1]:cq2, qf2p##2[k1]:cq2}, voltage = 1
```

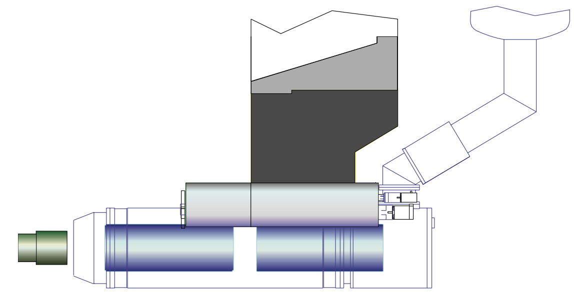
Example: Control room chromaticity tuning knob

```
xqune_1: group = {sex_08w:-6.4e-4, sex_09w:-1.9e-4, ...}, k2
```

Advantage of controller elements is that once defined in a lattice file, any program can read in the lattice and simulate the effect of the controller.

Cam Mover Controller Example

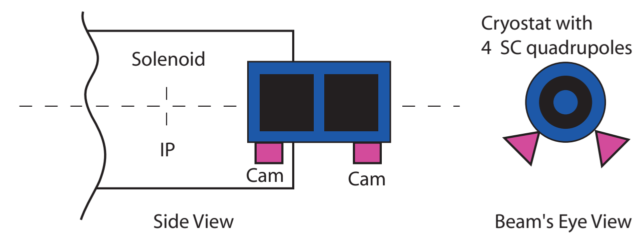
Controllers can control other controllers ad infinitum. Example: Cam movers supporting a superconducting cryostat which contains 4 spatially overlapping superconducting quadrupoles

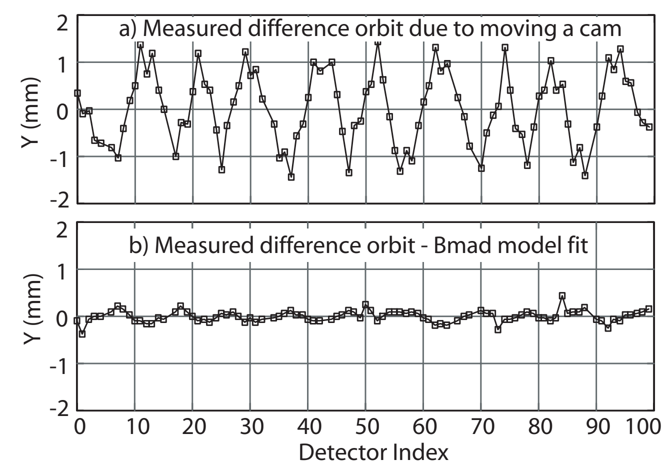


cam_1w: overlay = {tee_1w_x[x_offset]:cc_x,
tee_1w_y[y_offset]:cc_y}, rho

tee_1w_x: overlay = {q01w[x_pitch]:ds1,
q01w[x_offset]:dq1, q01w[tilt]:dt1,
q02w[x_pitch]:ds1, q02w[x_offset]:dq2,
q02w[tilt]:dt1, ...}, x_offset

tee_1w_y: overlay = {q01w[y_pitch]:ds1,
q01w[y_offset]:dq1_t1, q02w[y_pitch]:ds1,
q02w[y_offset]:dq2_t1, ...}, y_offset





Bmad, MAD, SAD Lattice Translation

MAD-8/MAD-X and Bmad:

- ~100% of a MAD lattice can be translated to Bmad.
- Not able to translate such things as controller elements, multipass, etc. from Bmad to MAD.
- MAD does not have a wiggler element but a Bmad wiggler can be approximately converted using a bend-drift-bend model.

SAD and Bmad:

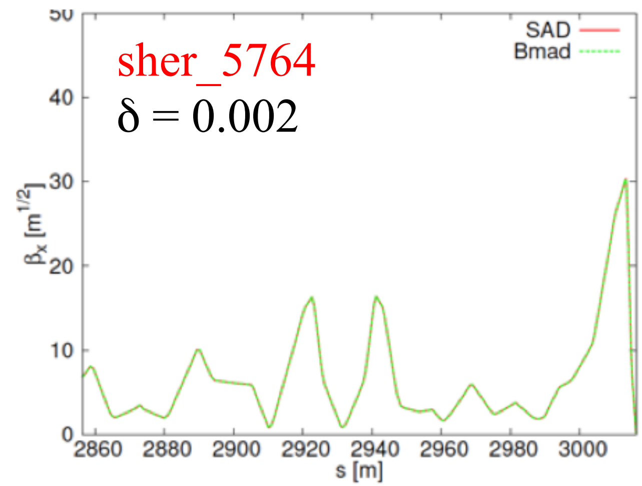
- In cooperation with Demin Zhou and Katsunobu Oide. Translation software from SAD to Bmad has been developed [New 2014!].
- Bmad to SAD translator under development

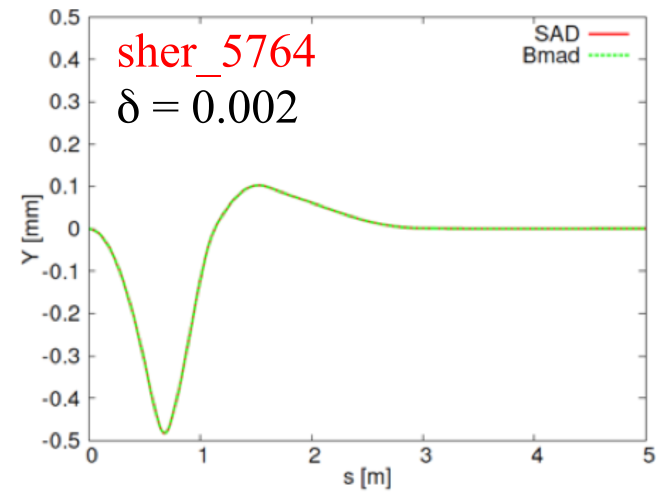
SAD:

```
mult   ecsle5  =(l =.01  f1 =.01 fringe =3 sk3=-1.18e-05);
```

Bmad:

```
ecsle5: sad_mult, f1 = 0.01, fringe_type = sad_full,
            l = 0.01, a3 = -1.18e-05 / factorial(3)
```



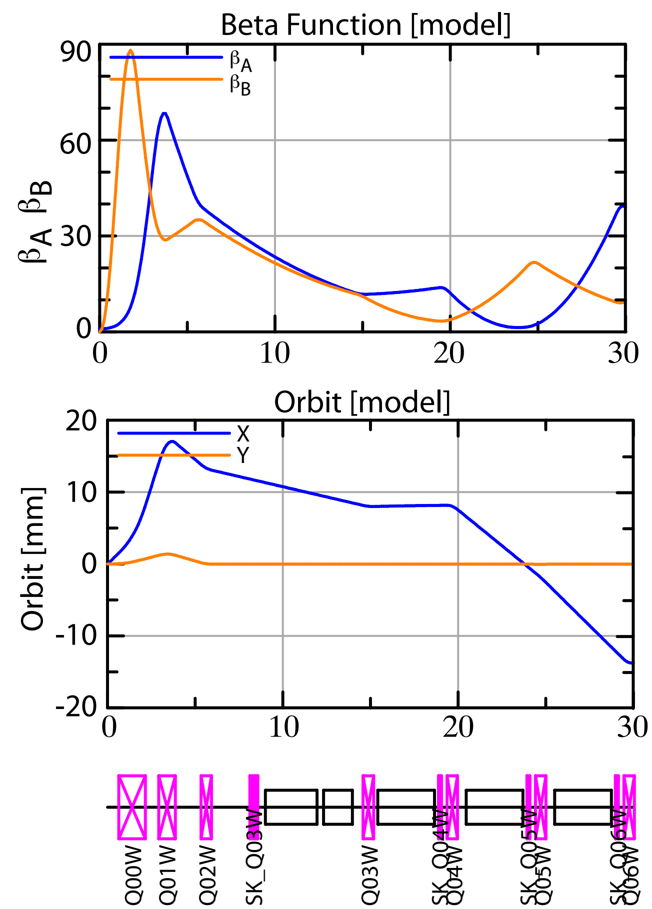


[Benchmark run by D. Zhou]

Bmad Ecosystem

Due to its flexibility, Bmad has been used in a number of programs including:

- **tao** General purpose design and simulation.
- **synrad3d** 3D tracking of synch photons, including reflections, within the beam chamber.
- **cesrv** On-line data taking, simulation, and machine correction for CESR.
- **dark_current_tracker** Dark current electron simulation.
- **freq_map** Frequency map analysis.
- **ibs_sim** Analytic intra-beam scattering (IBS) calculation.
- **touschek_track** Tracking of Touschek particles.
- etc...



Code reuse: Modules developed for one program can, via Bmad, be used in other programs.

Dark Current Tracker Program

Problem: Simulate in 3D dark current electrons generated at the walls of the beam chamber.

Challenges:

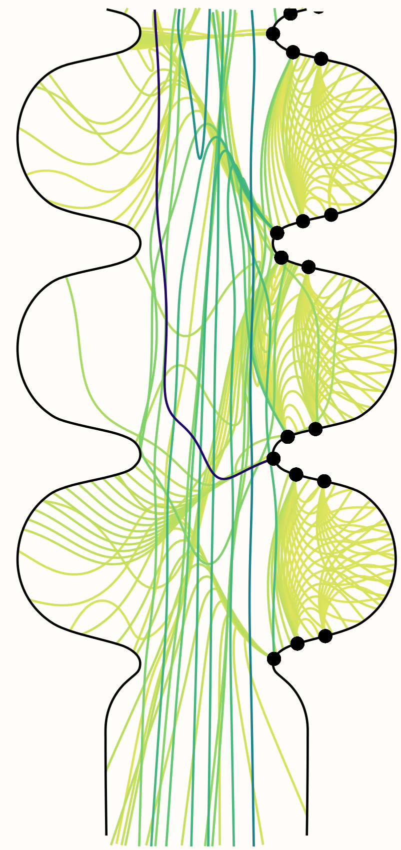
1. Define the beam chamber walls.
2. Be able to track particles that could reverse direction longitudinally.

Solutions: Develop in Bmad

1. X-ray capillary wall code extended for simulating beam chamber walls.
2. Time based tracker module that could handle bi-directional tracking.

Result: A useful program was developed and Bmad gets extended capabilities which have been used in other programs.

Dark current tracks in an RF cavity

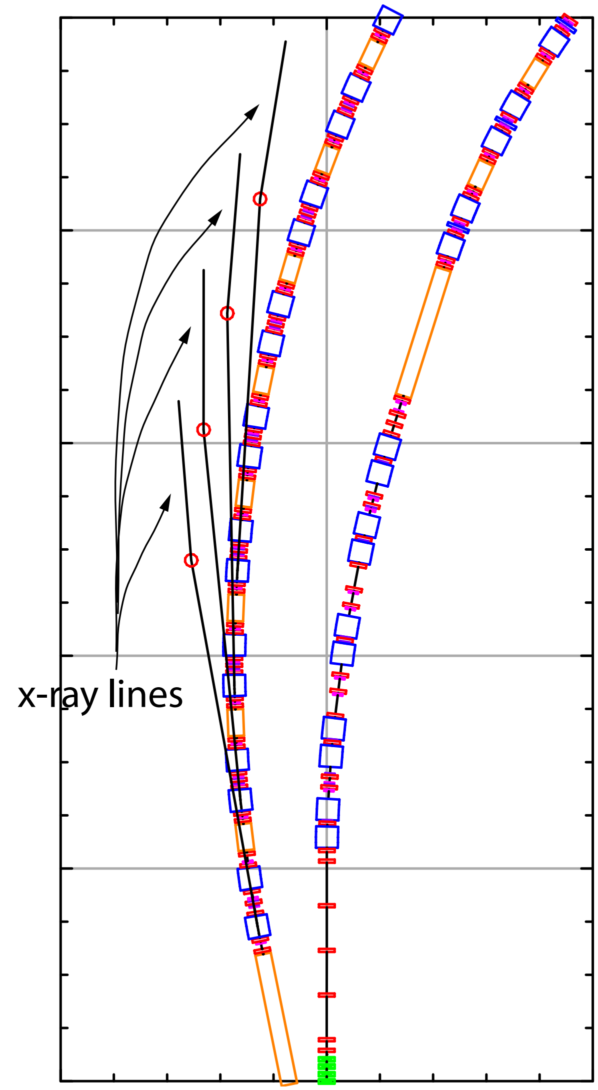


ERL Simulations

One area of current Bmad development is a unified ERL simulation framework.

Idea: To be able to simulate

- Electrons from the gun cathode to X-ray generation in wigglers and undulators through to the dump.
- X-rays from generation through to the experimental end stations.



Low Energy Simulations

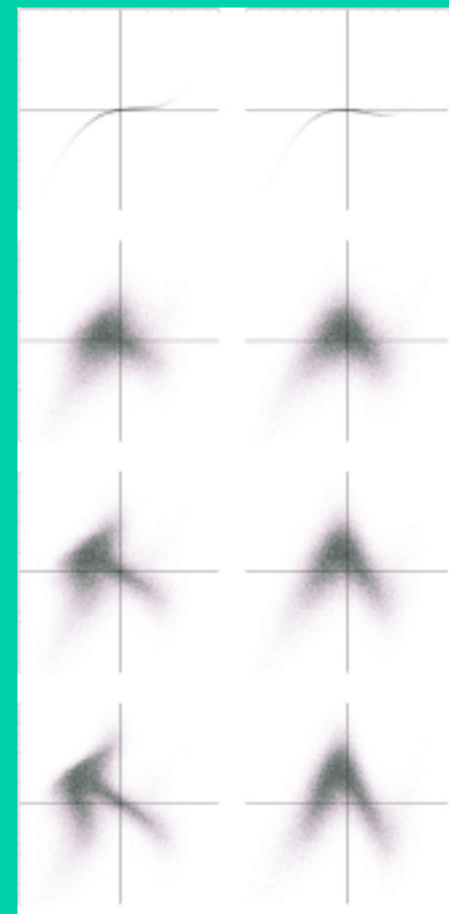
Have developed new lattice elements to handle low energy tracking

- **e_gun** Gun cathode region element.
- **em_field** General field element.

Status:

- Coherent synchrotron radiation (CSR) model implemented.
- Space Charge: Do not want to reinvent the wheel. Integration with existing SC codes ongoing:
  - OPAL (Andreas Adelmann)
  - Impact-T (Robert Ryne, Ji Qiang)

Cornell ERL Low Energy CSR Simulation

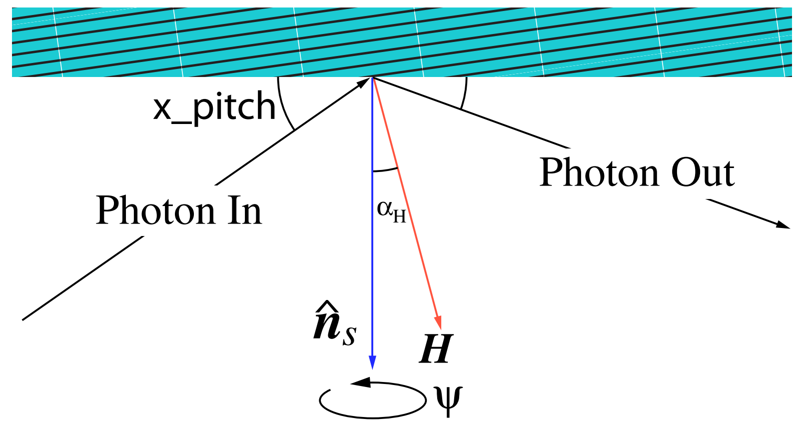


X-Ray Simulation

Both incoherent and coherent ray-tracing implemented (but needs more testing).

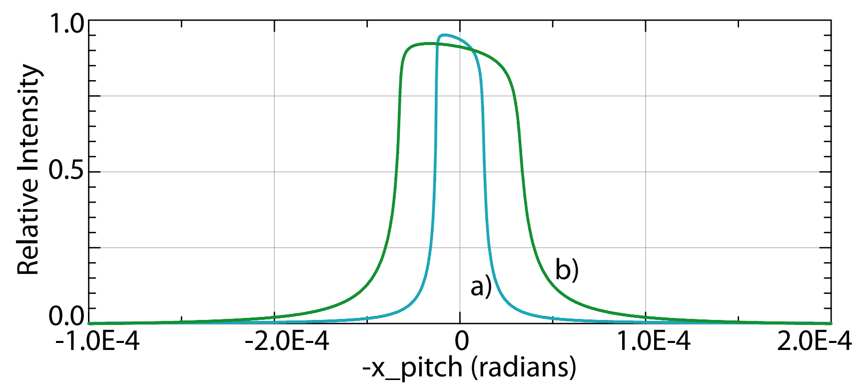
X-ray tracking elements developed:

- Crystal ! Bragg & Laue diffraction
- Capillary ! Focus X-rays
- Mirror
- Multilayer Mirror
- Diffraction_plate ! Apertures, Zone plates



Status:

- Bend, wiggler, and undulator incoherent X-ray generation implemented.
- Coherent X-ray generation from undulators under development.
- First "real world" simulations beginning.



Bragg crystal diffraction simulation

Synrad3D

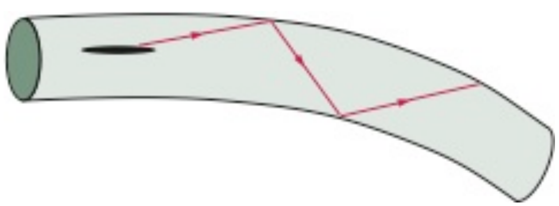
Synrad3D: Program to simulate in three dimensions:

1. Emission of photons from the beam
2. Tracking of photons to the vacuum chamber wall
3. Scattering or absorption at the chamber wall
4. If not absorbed, further tracking until photon is absorbed

Synrad3D was created to calculate the initial distribution of photoelectrons for electron cloud studies. It has also been used for studying the efficiency of beam masks.

At the start of development, Bmad provided code for:

- Calculation of the closed orbit
- Photon generation
- Defining the vacuum chamber wall
- If a photon is inside or outside the wall.



Synrad3d development involved:

- Tracking of photons.
- Reflection/absorption at the chamber wall.

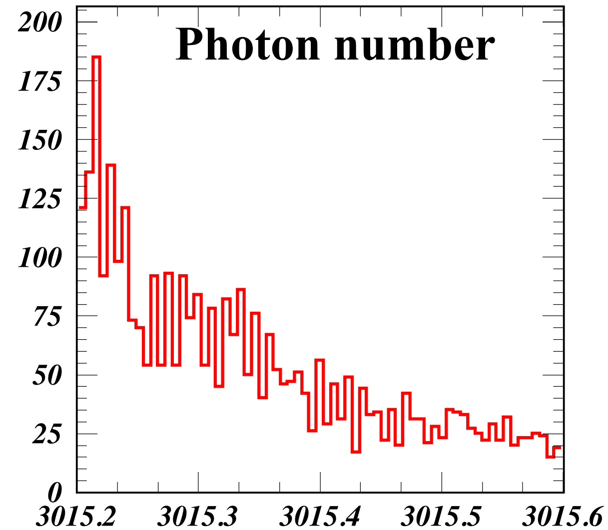
➔ The use of Bmad saved considerable development time.

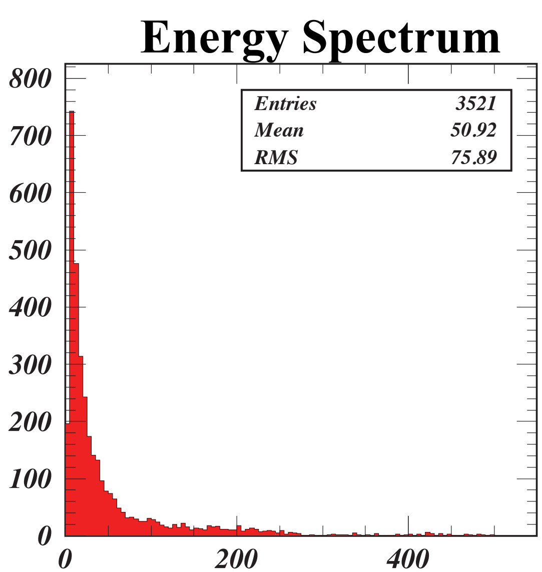
Synrad3d: SuperKEKB Simulation

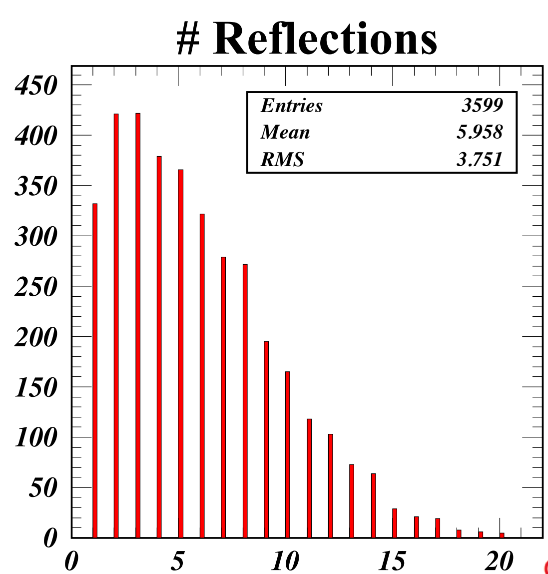
Simulations by Jim Crittenden
SuperKEKB wall profile courtesy Takuya Ishibashi

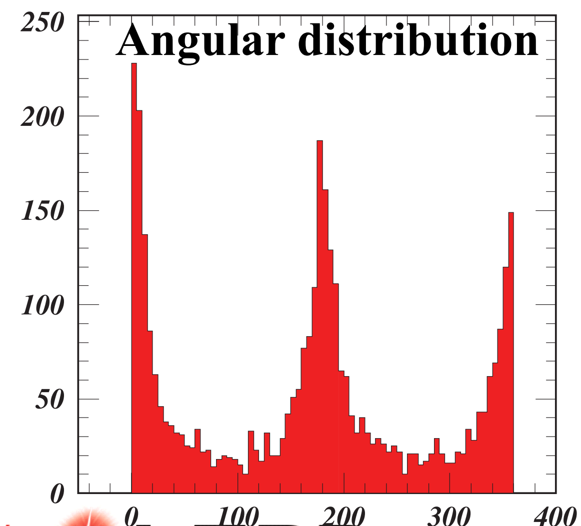
Problem: Photoelectrons can be trapped in the field of a quadrupole. The electrons will react with the beam and if the density is high enough, this can cause problems.

Simulation to determine the number of photons absorbed near a particular quadrupole









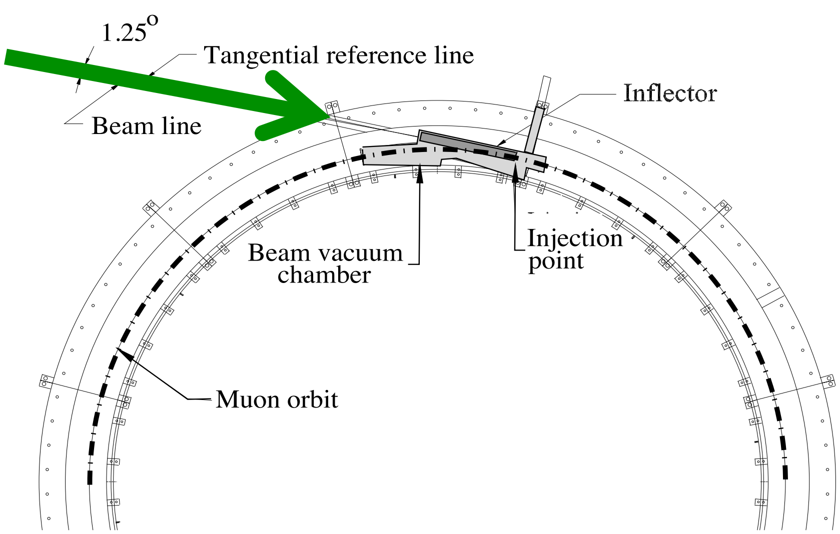
G-2 Experiment

Dave Rubin at Cornell has been developing a simulation program to simulate the Muon g-2 experiment at Fermilab

Need to simulate:

- Injection line into storage ring.
- Three dimensional field of the injection line.
- Scattering of muons as they cross the inflector wall
- Electrostatic quadrupoles
- Muon decay

Polarized Muons



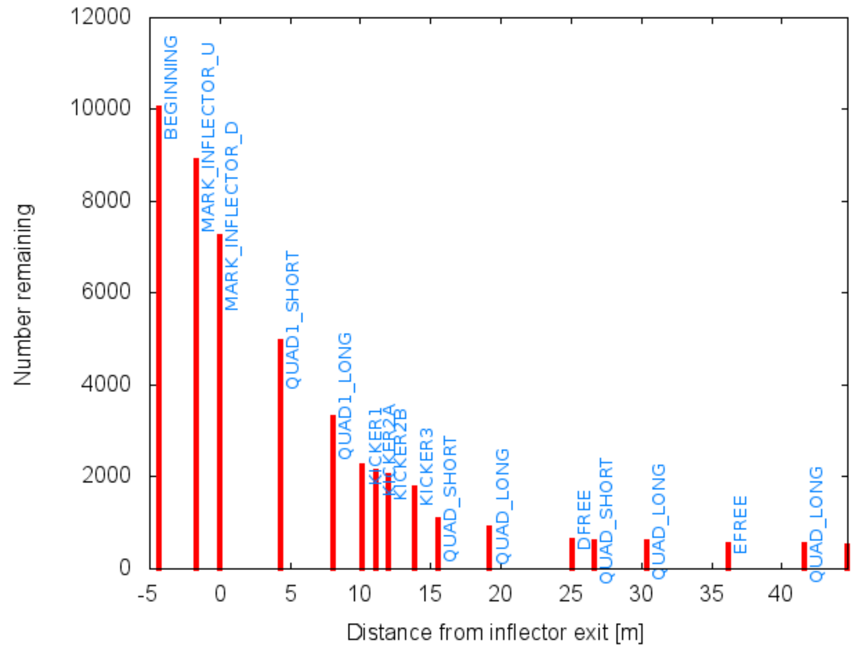
G-2 Simulation

Bmad provides:

Ability to define the geometry of the injection line and storage ring.
Ability to define the geometry of the inflector wall
Ability to define custom fields for the injection line and electrostatic quadrupoles

Needed to develop for the program:

Tracking of muons through the inflector wall
muon decay [will be ported to Bmad]



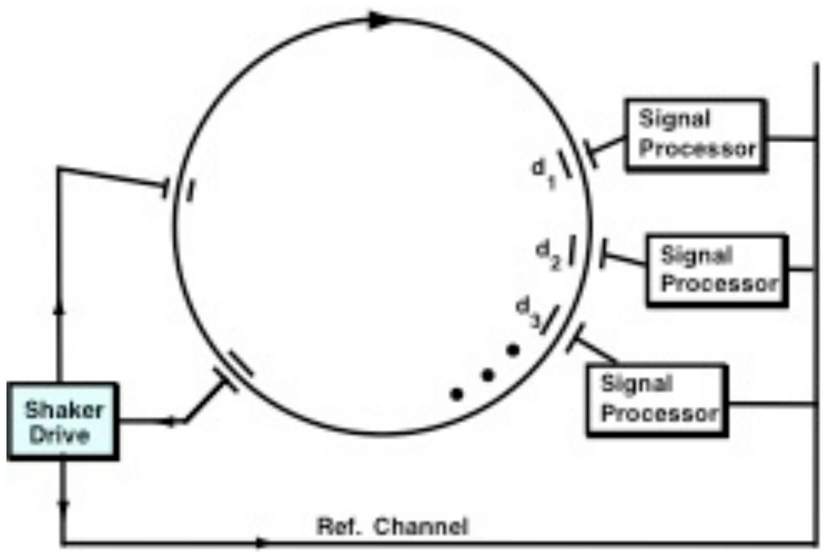
CesrV: Using Bmad in the Control System

CesrV is a online control system program for the e+/e- Cesr ring

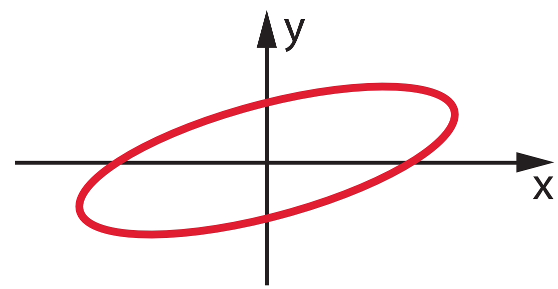
- Uses Bmad as its calculational engine
- Can take orbit, dispersion, betatron phase and coupling measurements
- Can correct orbits, dispersion, Twiss parameters, and coupling
- Can locate isolated steering, quadrupole, skew quadrupole errors.
- Can calibrate steerings, quadrupoles, skew quadrupoles.
- Etc. Etc.

CesrV: Twiss and Coupling Measurement

- Shake simultaneously both “a”-mode (horizontal like) and “b”-mode (vertical like) betatron resonance frequencies.
- About 100 detectors in Cesr
- Average over 40K turns
- Each detector has its own signal processor.
  Parallel measurement => Takes about 10 seconds



Measurement setup



Beam response at a detector due to “A”-mode (Horizontal like) shaking

CesrV: Coupling Analysis

Similarity transformation of 4x4 one turn map **T** at a given detector

$$\mathbf{T} = \mathbf{V}\mathbf{U}\mathbf{V}^{-1}$$

where **U** is block diagonal

$$\mathbf{U} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}$$

Write **V** in the form

$$\mathbf{V} = \begin{pmatrix} \gamma\mathbf{1} & \mathbf{C} \\ -\mathbf{C}^{+} & \gamma\mathbf{1} \end{pmatrix}$$

det|**V**| = 1 give condition

$$\gamma^2 + |\mathbf{C}| = 1$$

In terms of SAD R_2:

$$\mathbf{R}_2 = -\mathbf{C}^{+}$$

It is convenient to work with

$$\overline{\mathbf{C}} = \mathbf{G}_A \mathbf{C} \mathbf{G}_B^{-1}$$

Where

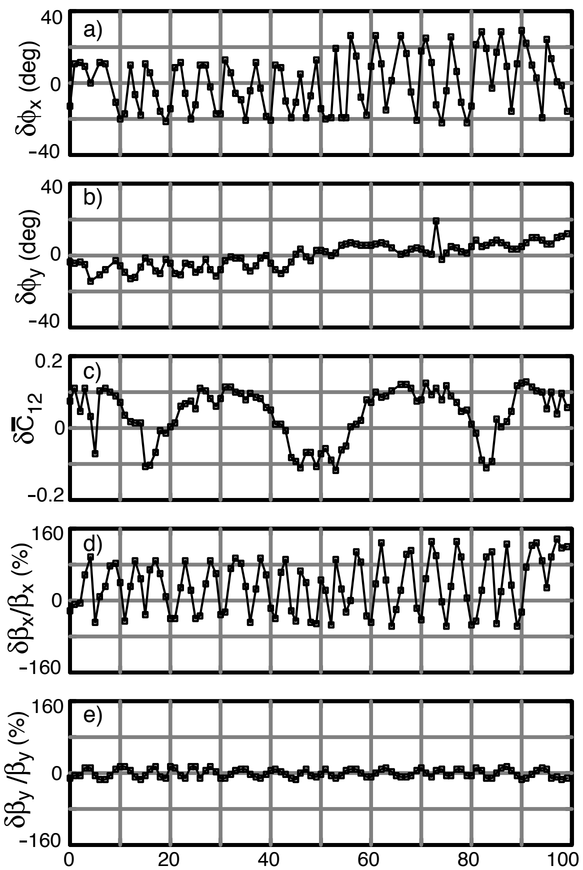
$$\mathbf{G} = \begin{pmatrix} 1/\sqrt{\beta} & 0 \\ \alpha/\sqrt{\beta} & \sqrt{\beta} \end{pmatrix}$$

$\overline{\mathbf{C}} = 0$ => Uncoupled motion

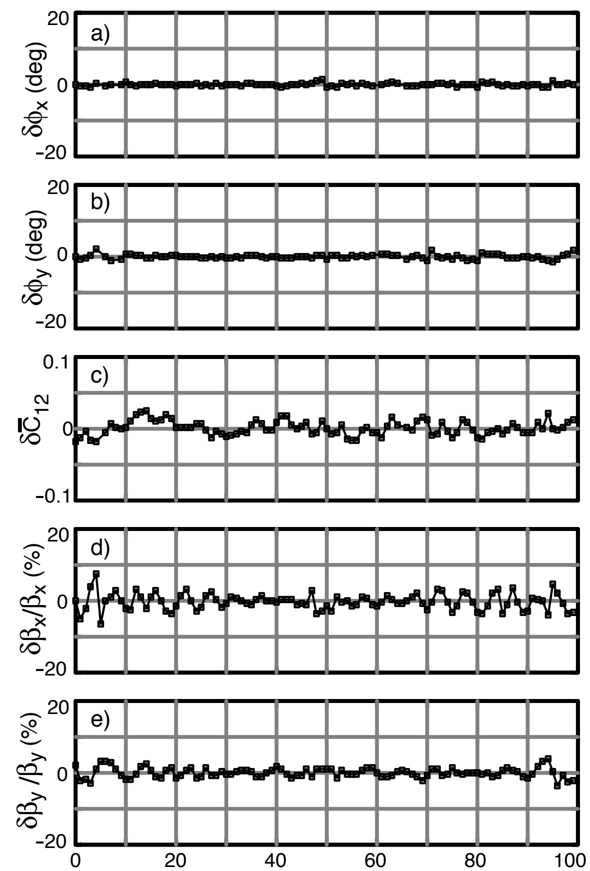
$\overline{\mathbf{C}} \sim 1$ => Fully coupled

CesrV: Twiss and Coupling Correction

Can measure: β_A, β_B, ϕ_A, ϕ_B, C_{11}, C_{12}, and C_{22}
In practice, cleanest is to use: ϕ_A, ϕ_B, C_{12}
Correction calculation takes less than 1 minute

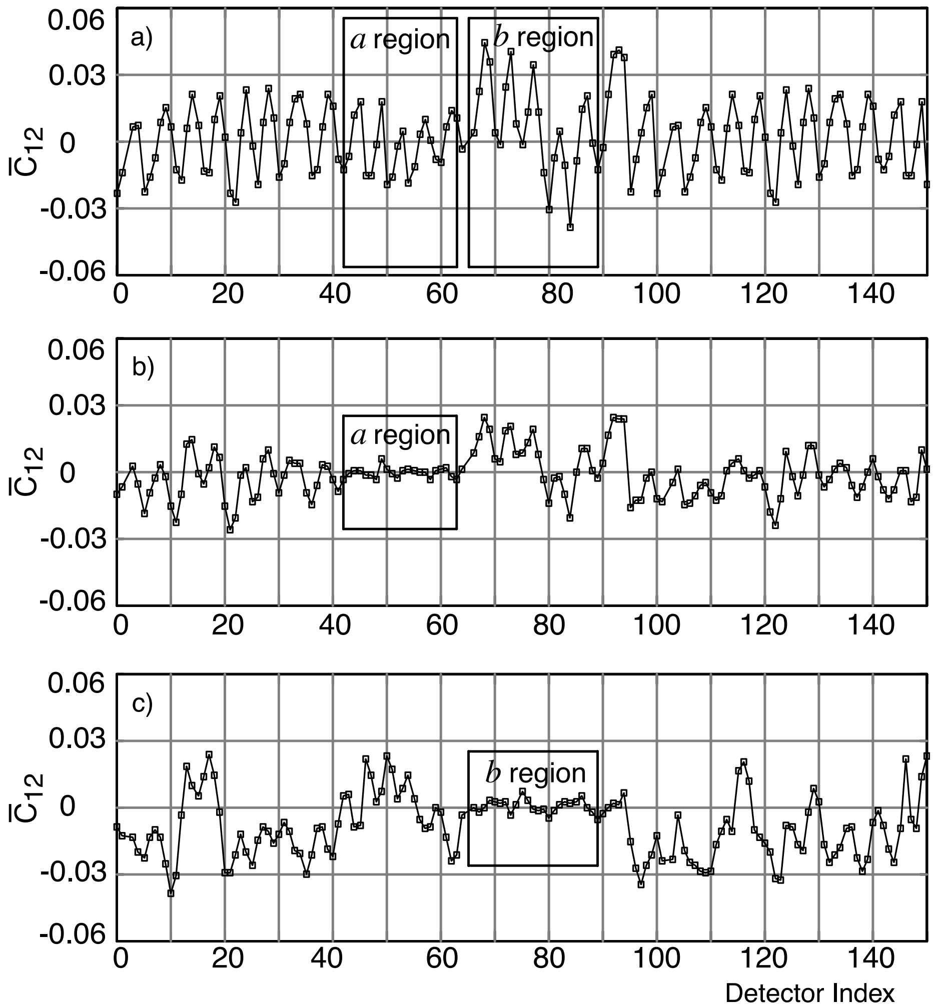


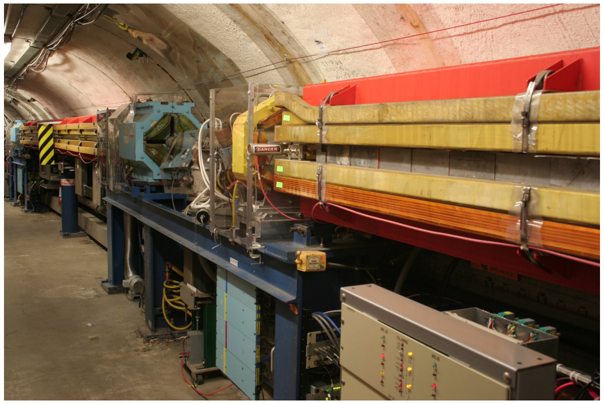
Before correction



After correction

Locating a Skew Quadrupole Error





The Future

Bmad has evolved continually since its birth and shows no sign of slowing down.

Current projects include:

- SAD compatibility development.
- Further integration with PTC
- Undulator coherent X-ray generation.
- Testing of partially coherent X-ray tracking.
- Nonlinear controllers
- ...

Conclusion

- Bmad has been used successfully at Cornell for a number of years.
- With Bmad, new simulation programs can be developed in less time and with less effort. [Caveat: Learning to program with Bmad has a significant learning curve.]
- Bmad is constantly evolving to meet changing needs.
- Collaborations welcome.