The Bmad Software Toolkit as a Framework for Simulation development

David Sagan
Cornell Laboratory for Accelerator-Based Sciences and Education
Brief History:

• Bmad is a software toolkit for the simulation of charged particles and X-rays.
• Born at Cornell in mid 1990’s
• Started life as modest project: Just wanted to calculate Twiss functions and closed orbits.
• 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:
- Lattice design
- X-ray simulations
- Spin tracking
- Wakefields and HOMs
- Beam breakup simulations in ERLs
- Intra-beam scattering (IBS) simulations
- Coherent Synchrotron Radiation (CSR)
- Touschek Simulations
- Frequency map analysis
- Dark current tracking
- Etc., etc.
Overview

- Written in Fortran 2008.
- Object oriented from the ground up.
- Has structure translation code for interfacing with C++.
- With certain restrictions, Bmad can be run multi-threaded.
- Lattice files use a MAD like syntax.
- Well documented (Manual is ~500 pages).
- Open Source: http://www.lepp.cornell.edu/~dcs/bmad/

```fortran
type (lat_struct) lat

call bmad_parser ('lat.bmad', lat)
```
Advantages of a toolkit:

• Cuts down on the time needed to develop programs.
• Cuts down on programming errors (via code reuse).
• Provides a simple mechanism for lattice function calculations from within control system programs.
• Standardizes sharing of lattice information between programs.
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.
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 (like MAD) with
- Twiss and orbit calculations.
- Nonlinear optimization.
- Spin tracking.
- Etc.

Additionally: Tao’s object oriented coding makes it relatively easy to extend it.
- For example: For Cornell/BNL’s CBETA ERL/FFAG machine, will use an extended Tao as an online model for orbit flattening, etc.
  [Need < 1K lines of code!]

Tao with Bmad gives the flexibility of a library with the convenience of a program.
Bmad has a number of features that over the years have proven useful...

Among these are:

- Define overlapping elements.
- Join beam lines together.
  [EG: injection line connecting to a ring.]
- Beamlines sharing common elements
  [EG: The IR region in a dual ring colliding beam machine.]
- Can define vacuum chamber walls.
- Custom elements and custom particle tracking
- Can construct Taylor maps for normal form analysis.
- Controllers – Elements controlling attributes of other elements.
  [EG: Simulate a control system.]
Low Energy Simulations

Have lattice elements to handle low energy tracking:

• e-gun Gun cathode region element.
• em_field General field element.

Status:

• 1D/3D Coherent Synchrotron Radiation (CSR) model implemented.
• Very low energy space charge: Do not want to reinvent the wheel. Integration with existing SC codes ongoing:
  • OPAL (Andreas Adelmann)
  • ASTRA
  • GPT
Spin Tracking

- Bmad and PTC can track a particle’s spin including EDM and fringe fields.
- Bmad can track through arbitrary fields.
- Can produce transfer maps which include spin.

Phase Space Map

<table>
<thead>
<tr>
<th>Out</th>
<th>Coef</th>
<th>Exponents</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>-0.600000000000</td>
<td>0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1:</td>
<td>1.000000000000</td>
<td>1 0 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1:</td>
<td>0.145000000000</td>
<td>2 0 0 0 0 0</td>
<td>2</td>
</tr>
<tr>
<td>2:</td>
<td>-0.185000000000</td>
<td>0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>2:</td>
<td>1.300000000000</td>
<td>0 1 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>2:</td>
<td>3.800000000000</td>
<td>2 0 0 0 0 1</td>
<td>3</td>
</tr>
<tr>
<td>3:</td>
<td>1.000000000000</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>4:</td>
<td>1.000000000000</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>5:</td>
<td>0.000000000000</td>
<td>0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>5:</td>
<td>0.000001480008</td>
<td>0 1 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>5:</td>
<td>1.000000000000</td>
<td>0 0 0 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>5:</td>
<td>0.000000000003</td>
<td>0 0 0 0 1 0</td>
<td>1</td>
</tr>
<tr>
<td>5:</td>
<td>0.000000000003</td>
<td>2 0 0 0 0 0</td>
<td>2</td>
</tr>
<tr>
<td>6:</td>
<td>1.000000000000</td>
<td>0 0 0 0 0 1</td>
<td>1</td>
</tr>
</tbody>
</table>

Spin Map

<table>
<thead>
<tr>
<th>Out</th>
<th>Coef_Sx</th>
<th>Coef_Sy</th>
<th>Coef_Sz</th>
<th>Exponents</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sx:</td>
<td>0.99757886</td>
<td>-0.02372254</td>
<td>-0.06537314</td>
<td>0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Sx:</td>
<td>0.04802411</td>
<td>0.22401654</td>
<td>0.6515483</td>
<td>1 0 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sx:</td>
<td>0.07391383</td>
<td>3.77388795</td>
<td>-0.24137562</td>
<td>0 1 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sx:</td>
<td>0.00008802</td>
<td>-0.17584738</td>
<td>0.06515458</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sx:</td>
<td>0.01148244</td>
<td>-0.20322945</td>
<td>0.24896717</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sx:</td>
<td>-0.00457076</td>
<td>-0.22322148</td>
<td>0.01125358</td>
<td>0 0 0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>Sy:</td>
<td>-0.02460178</td>
<td>0.75885811</td>
<td>-0.65079114</td>
<td>0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Sy:</td>
<td>0.25039365</td>
<td>-0.04801091</td>
<td>-0.06544895</td>
<td>1 0 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sy:</td>
<td>-3.02631629</td>
<td>-0.07739091</td>
<td>0.02416143</td>
<td>0 1 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sy:</td>
<td>0.17584738</td>
<td>0.00008802</td>
<td>-0.00654489</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sy:</td>
<td>0.47579058</td>
<td>1.59361755</td>
<td>1.84025908</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sy:</td>
<td>0.13046518</td>
<td>-0.46155512</td>
<td>0.54313051</td>
<td>0 0 0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>Sz:</td>
<td>0.06504736</td>
<td>0.65082378</td>
<td>0.75643719</td>
<td>0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Sz:</td>
<td>-0.64180483</td>
<td>0.06414595</td>
<td>0.00000000</td>
<td>1 0 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sz:</td>
<td>-2.27815007</td>
<td>0.22777762</td>
<td>-0.00007328</td>
<td>0 1 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sz:</td>
<td>0.06515785</td>
<td>-0.00651227</td>
<td>0.00000000</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sz:</td>
<td>0.00385332</td>
<td>-1.86555986</td>
<td>1.60475990</td>
<td>0 0 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Sz:</td>
<td>0.11944181</td>
<td>0.53003513</td>
<td>-0.46630288</td>
<td>0 0 0 0 0 1</td>
<td>1</td>
</tr>
</tbody>
</table>
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
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.
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

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

Energy Spectrum

Number of photons absorbed near a particular quadrupole

Energy Spectrum

Mean: 50.92
RMS: 75.89

Entries: 3521

# Reflections

Mean: 5.958
RMS: 3.751

Entries: 3599

Angular distribution

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

Need to track the spin of polarized muons with:

- Injection line into a 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 quads

Needed to develop for the program:
Tracking of muons through the inflector wall
muon decay [will be ported to Bmad]
Bmad is being developed on many fronts including:

• Integration with the **Genesis** FEL simulation program for use with stochastic optical cooling simulation.

• **Fast Spin tracking simulation** development (in conjunction with Jlab electron-ion collider design).

• Integration of **low energy space charge** developed by Rob Ryne directly with Bmad (Chris Mayes).

• Using Taylor maps and normal form analysis to develop a **fast dynamic aperture analysis** program (Michael Ehrlichman).
Conclusion

• Bmad is an open source software library for simulating charged particle beams and X-rays.

• With Bmad, new simulation programs can be developed in less time and with less effort and with fewer bugs.

• Bmad has been successful due to it’s modular, object-oriented design which allows it to be adapted to ever changing simulation needs.

• Has been used at Cornell, CERN, KEK, DESY, ANL, BNL, Cockcroft Institute, for simulating CESR, ILC, ATF2, SuperKEK-B, Cornell ERL, g-2 muon project, etc.

• Bmad comes with an ecosystem of programs for lattice design, dynamic aperture calculations, Touschek simulation, etc., etc.

• Bmad can serve as a framework for start-to-end simulations.

• Bmad is always in continual development with about 2 FTEs.

• Supported by Cornell Laboratory for Accelerator-based Sciences and Education (CLASSE).
Questions?