Recent Improvements to CHEF

Jean-Francois Ostiguy*
Leo Michelotti

PAC 2009, Vancouver
What is CHEF?

- a body of software for accelerator computations, a set of libraries and a standalone application. The application uses a subset of the capabilities provided by the libraries which are meant to be used independently.
- native C++ implementation
- Ancestry dates back to the early 1990s
Late 1980s and 1990’s:

- A. Dragt, J. Irwin, E. Forest and others show how Hamiltonian dynamics in the context of accelerators can be formulated in a way that puts (local) maps, rather than (global) Hamiltonians in the central role.

- Analytical maps can be approximated by Taylor series. M. Berz shows that automatic differentiation techniques are a natural fit to perform operations on Taylor maps and releases the first production quality AD engine (the basis for the code COSY).
Design Objectives

• Unified framework for:
  ➢ tracking and map generation
  ➢ linear and nonlinear analysis

• Approximations under user control

• Separated propagation physics

• Local geometry
Library Hierarchy

components

High level Components

- Site viewer
- Beamline browser
- Phase space display etc ...

physics_toolkit

Tools for calculating lattice functions, moment propagation, normal form analysis etc ...

YACC based parsers. MAD8, XSIF, MADX

bmlfactory

Beamline creation, manipulation and edition. Tracking.

Mxyzptlk

Automatic differentiation/differential algebra engine for perturbation theory and tracking.

basic_toolkit

Utility functions, Linear Algebra, Memory management, Physical Constants
Automatic Differentiation

Consider $f$ and its derivatives evaluated at $x_0$. For any composition $g = h \circ f$ one can compute $g$ and its derivatives at $x_0$ exactly.

Algebra of derivatives $\iff$ Algebra of polynomials

Performance and minimal memory footprint are the main challenges

The no of coefficients is large: $(n + m)! / n! \cdot m!$

Implementation requires careful attention to low level details:
memory management, individual coefficient access mechanism, etc
**Jet**: a data type that aggregates all coefficients. Operations are generalized to this type (similar to what is done for matrices). Other types (Maps, LieOperator use Jet as a building block.

- Takes max advantage of C++ operator overloading
- Templates and implicit conversion provide simultaneous support for complex and double Jet types. The latter can be mixed.
- Reference point and space dimensionality are a run-time choice.
- Highly optimized implementation:
  - 6\(^{th}\) order (6d) map for a ring with a few 100 elements can be computed in >10 min
  - At first order, overhead compared with traditional matrices is minimal.
  - Small and dynamic memory footprint
Hierarchical (single rooted tree)
• container endowed with standard STL iterators (depth or breath first)
• can be edited, transformed

Parsers
• based on formal (bison/yacc) GLR grammar
• MAD8, XSIF, MADX(soon)
  • expressions
  • macros
• hierarchical structure is preserved in constructed model
Geometry

- Propagation physics expressed in “natural” local coordinates
- “empty space” handles frame transformations between elements, allowing arbitrary placement
- general misalignments may be handled exactly using this mechanism
- alternatively, small misalignments may be handled approximately and locally
- (sub) beamlines may be misaligned as a whole, emulating the behavior of rigid girders
Propagator Architecture

Beamline Element

- Propagation physics is encapsulated in a Propagator object
- Propagators can be specified at runtime
- Useful to introduce domain-specific approximations
- Propagator handles finite aperture extent when specified (implemented as decorator)
- Details of propagation (e.g. thin kicks, integrator) are hidden from the rest of the code
- Propagators propagates either Particles or JetParticles (templated)
Analysis

- At first order, Maps obviously allow standard lattice computations, coupled and uncoupled.
- Generalization to higher orders is also available: normal forms, higher order dispersion, momentum compaction factor and so forth.
Lattice Functions

Beamlines (Linac)  Ring

Standard and coupled functions for both lines and rings.
Choice of parametrizations for coupled lattice functions.
Normal Forms

- Phase space contours for a sinusoidal rf bucket in std (E-phi) coordinates (top left)
- After successive Normal Form transformations at orders 1, 3 and 7
Embedded Database

- Embedded SQL database is used internally to store and manage computed quantities (lattice functions, ref orbit, maps, labels etc)
- Standard queries enable element selection, tabulation, sorting etc
- Established tools are available to browse and display exported data.
Python Bindings

- Many codes are now structured as a high level control shell, implemented in an interpreted language. Computation intensive functionality is implemented in a compiled language and accessed through a dedicated layer.
- CHEF provides a comprehensive set of bindings for python, which is widely used and is a good match for c++ concepts.
- The code SYNERGIA (3d space charge) uses this mechanism to import needed functionality.
**Script Examples**

\[ \sqrt{x+y+z+px+py+pz} \quad \text{Order} = 12 \]
Linac Modeling

Although maps techniques remain useful for single pass machines (e.g. final focus design), the emphasis in that context is more on conventional tracking.

CHEF has been adapted for high energy linacs. Specifically, we added:
- reference trajectory in the presence of acceleration
- Linac-specific accelerating structure element
- wakefields

Comparisons with other linac codes show excellent agreement.
Benchmark: Trajectory w/misalignments

Sample trajectory in ILC linac with misaligned elements (cavities and quads).

Max vertical scale: 1 mm

Comparison with Lucretia (SLAC)

Difference trajectory is shown.

Max vertical scale: 0.4 microns
Benchmark: Vertical Emittance Evolution in ILC

Comparison with Lucretia (SLAC)

Vertical emittance
Wakefields OFF
(same misalignments + correctors set to value determined by an emittance minimization algorithm)

Vertical emittance
Wakefields ON
Current Development

- Named “Knobs” and “measurements” objects based on generalized function objects (closures)
  Can be an element attribute, or a map coefficient or an arbitrary expression involving others
  (A proof-of-principle implementation exists)

- A basic localized space charge element (based on rigid gaussian / uniform distributions). The sc distribution is determined from the (unperturbed) beam moments.
Of Interest

**Synergia**

TH5PFP018
Recent Advances in the Synergia Accelerator Simulation Framework
J. F. Amundson, P. Spentzouris, E. G. Stern (Fermilab)

TH5PFP017
Space Charge Simulations for the Mu2e Experiment at Fermilab
J. F. Amundson, P. Spentzouris, E. G. Stern (Fermilab)

**COSY**

WE3PBC05
Advanced Simulation and Optimization Tools for Dynamic Aperture of Non-Scaling FFAGs and Related Accelerators including Modern User Interfaces
C. Johnstone (Fermilab) M. Berz, K. Makino (MSU) P. Snopok (St. Petersburg State University)

**MADX/PTC**

TH6PFP081
Resonance Driving Term Experiment at DAFNE
C. Milardi (INFN/LNF) F. Schmidt (CERN)