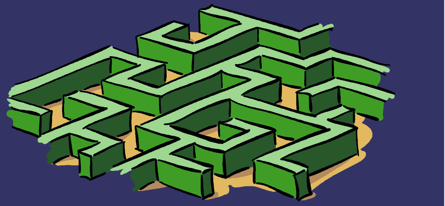
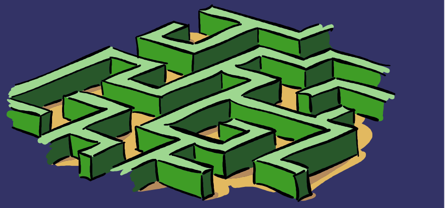
ATF2 Flight Simulator & Controls Interface

Glen White
SLAC
October 2007



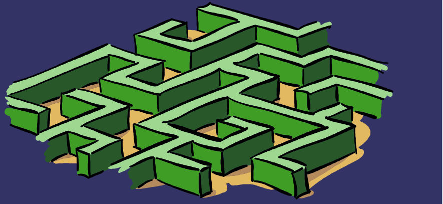
Overview

- Online ATF2 modeling
  - Lucretia & AML
- Online implementation of beam code (e.g. Tuning algorithm, BBA etc.)
  - How to develop and test beam dynamics code offsite and then implement code at ATF2?



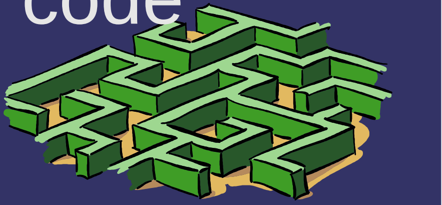
Online ATF2 Model

- At SLAC, Matlab + Lucretia heavily used for constructing beam dynamics code, tuning algorithms etc.
- As Lucretia is embedded into Matlab, very flexible
  - Use BEAMLINE description to store all required ATF2 info
  - Online Matlab code talks to EPICS and/or V-System to maintain up-to-date model parameters.
  - Plan to mirror this in AML for non-Matlab users.



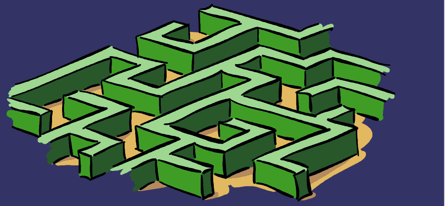
Accelerator Markup Language (AML)

- Use of any particular modeling code may force others into using the same code.
- The AML project aims to develop a universal accelerator format.
  - Based on XML so easy to extend.
  - Developing a Universal Accelerator Parser (UAP) to facilitate parsing of AML lattice into another form.
- AML <-> MAD parser already written.
  - UAP makes writing of parsers for other code simple.



ATF2 Code Integration

- Need a way to easily facilitate offsite writing and testing of code to be run at ATF2.
- NB: Remote running of code external to ATF control room forbidden.
- Currently discussing best way to do this
  - Through flight simulator (Lucretia + EPICS)
  - Directly through V-system API



Code Implementation Through Flight Simulator

- One possible mechanism:
  - Lucretia can access EPICS variables through Matlab using MML (Matlab Middle-Layer) interface.
  - Flight simulator software can be downloaded to home institute along with current machine description.
  - Write (Matlab) code to implement desired algorithms etc and test on flight simulator locally.
  - Take code to ATF and it should run in the same fashion on the realtime software.
  - Equivalent process possible to set up through AML.