Status of loss maps simulations with Merlin

J. Molson, R. Appleby, R. Barlow, A. Toader, M. Serluca

University of Manchester, Cockcroft Institute, University of Huddersfield

27th July, 2012
What is Merlin?

- C++ Accelerator physics library
- Provides a set of useful functions for accelerator modelling
- Initially used to simulate ground motion in the ILC BDS and linac
- Later the ILC damping rings
- Written by Nick Walker et al (DESY)
- Now adapted for large scale proton collimation simulations by Manchester and Huddersfield
- Three main sections of the library
  - Accelerator lattice loading/creation and storage
  - Tracker
  - Physics processes
- Modular design
Accelerator lattice creation

- Can load directly from MAD (tfs table output)
- Can also use XTFF format
- Direct element addition
- The created AcceleratorModel element can be further manipulated in the future, e.g. adjust aperture, alignment errors, etc
- AcceleratorComponent: The base class for each element in the lattice that all elements inherit from.
- EMField: The field associated with the element
- AcceleratorGeometry: Any Geometry transforms for the element, e.g. tilt
- Aperture: The aperture for the element, e.g. the beam pipe or collimator jaws
- WakePotentials: Any wakes for the element - resistive wall, geometric and cavity wakes
Tracker

- Different types of tracker, particle tracking and moment tracking
- Takes the input of a bunch and beamline, and tracks the bunch along the beamline
- Can specific integrator sets, e.g. transport, thin lens, symplectic
- Can override specific integrators, e.g. crab cavities
- Step both along the accelerator lattice and within individual elements
Physics processes

- Additional physics on top of tracking to be applied at selected elements and positions
- Can be enabled or disabled as required - processes are attached to trackers
- Examples: Synchrotron radiation, collimation, wakefields, etc
- Easy to create, template examples exist
- Trackers manage stepping within processes - inputs are the AcceleratorComponent and bunch
Accelerator errors

- Can offset element positions, x,y,z
- Can adjust angular tilts
- Can add in field errors including additional multipoles
- Can generate errors inside merlin
- We generate errors in MAD, correct for errors, then transfer this information to merlin
- Tested loss maps with an errored + corrected lattice, with collimators aligned to the perfect orbit
- Little difference from the perfect configuration in loss maps
Parallel running

- Wish to run large simulations - very cpu heavy - use MPI
- Must use multiple physical machines with interconnects
- Run multiple copies of the same binary that can communicate with each other
- Tracking, collimation, etc, are all independent on a per-particle basis, do not need any knowledge about other particles
- Collective effects such as space charge, wakefield, etc do require this information
- Functions exist such as parallel bunch moment calculations (mean, standard deviation) in addition to the ability to move particles between computers
- Parallel running is implemented at a per process algorithm level
Parallel running

MPI::Init()  \rightarrow \text{Bunch Creation}  \rightarrow \text{Linear Tracking}  \rightarrow \text{Collective effects}  \rightarrow \text{Linear Tracking}  \rightarrow \text{MPI::Finalize()}

- Node 1
- Node 2
- Master
- Node n-1
- Node n
Currently working on parallelising processes that contain collective effects
e.g. Wakefield + Space charge effects
This is Algorithm dependent.
Collimation simulation configuration

- Want to have a comparison with Sixtrack
- Thick-lens version V6.5.2012.02.seq
- Using Beam 1
- $\beta^*$ for IP1 and IP5: 0.6m
- $\beta^*$ for IP2 and IP8: 3m
- 6.4 M particles simulated
- No field or alignment errors
- Energy= 4 TeV, $\epsilon_n = 3.5$ mm-mrad, $dp/p = 0$, $\sigma_z = 0$
- Crossing angle[$\mu$rad]: X1=-145, X2=-90, X5=145, X8=-220
- Parallel separation on at all IP: sep = $\pm 0.65$mm
- Horizontal halo cut at $4.3\sigma$, similar results with different halo distributions
- Impact parameter = 1$\mu$m and 10 cm longitudinal loss resolution
Collimation simulation configuration

- Beam injected at TCP.C6L7.B1
- Collimators aligned to orbit and beam envelope

<table>
<thead>
<tr>
<th>Collimator</th>
<th>Aperture (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP IR7</td>
<td>4.3</td>
</tr>
<tr>
<td>TCP IR3</td>
<td>12</td>
</tr>
<tr>
<td>TCSG IR7</td>
<td>6.3</td>
</tr>
<tr>
<td>TCSG IR3</td>
<td>15.6</td>
</tr>
<tr>
<td>TCLA IR7</td>
<td>8.3</td>
</tr>
<tr>
<td>TCLA IR3</td>
<td>17.6</td>
</tr>
<tr>
<td>TCLP</td>
<td>10</td>
</tr>
<tr>
<td>TCT IR1/IR5</td>
<td>9</td>
</tr>
<tr>
<td>TCT IR2/8</td>
<td>12</td>
</tr>
<tr>
<td>TCDQ IR6</td>
<td>7.6</td>
</tr>
<tr>
<td>TCLI</td>
<td>open</td>
</tr>
<tr>
<td>TDI</td>
<td>open</td>
</tr>
</tbody>
</table>

J. Molson, R. Appleby, R. Barlow, A. Toader, M. Serluca

ColUSM #10 Status of loss maps simulations with Merlin 11/28
Beta functions

MAD-X vs. Merlin Optics

Merlin $\beta_x$
MADX $\beta_x$
Beta functions - IR5

![Graph showing beta functions comparison between Merlin and MADX](image-url)
Beta functions - IR7

![Graph showing beta functions for IR7](image)
Reference orbit

![Graph showing orbit in reference frame](image)
Reference orbit - zoom x

![Graph showing reference orbit zoomed in]
Loss map results comparison (Sixtrack plots from R. Bruce talk)
Loss map Results IR7

MERLIN

SIXTRACK

J. Molson, R. Appleby, R. Barlow, A. Toader, M. Serluca

ColUSM #10 Status of loss maps simulations with Merlin 19/28
Have been working on enhanced scattering physics inside a collimator jaw

- Nuclear interactions - pA scattering
- Nucleon scattering: elastic and single diffractive
- Higher level electron interactions - multiple coulomb scattering, atomic ionization
- Do not care about any secondary particles, similar to sixtrack
- Aiming to be precise and fast
"Our results for the final-state particle distributions are remarkably insensitive to the mass of the target nucleus. Mean pseudorapidity, mean rapidity, and multiplicity, plotted as a function of $(1 - x_f)$, show the same behaviour for Be, Al, and W. This strongly suggests that the dominant process is the diffractive excitation of single nucleons, an observation which is supported by the $A^{1/3}$ dependence of the diffraction cross-section."

Diffraction dissociation of nuclei in 450 GeV/c proton-nucleus collisions (Z Phys C, V49, 1991)

"These results indicate that target diffraction occurs on only one single nucleon of the nucleus. However, the abundance of positive particles emitted in the backward direction indicates that there must be re-scattering of the emitted particles on other nucleons of the nucleus."

Nuclear-target diffraction dissociation in $\pi^+$ and $K^+$ collisions with Au and Al at 250 GeV/c (Z Phys C, V72, 1996)
Regge theory

- Older theory, but describes soft interactions
- Introduce the idea of complex angular momentum
- Result: cross sections $\sigma \sim s^{(L-1)}$
- Reggeons: mesons ($a, f, \rho, \omega$) - results in falling cross sections
- Pomerons: thought to be glueballs - results in rising cross sections
- Our range of interest is of beam energy between 450 and 7000 GeV/c
- This is $\sqrt{s} = 30 \rightarrow 115$ GeV for fixed target interactions
Elastic scattering

- Interested in the differential cross section $\frac{d\sigma}{dt}$
- Fit all appropriate existing $pp$ and $p\bar{p}$ data
- Data exists on either side of the region of interest so interpolation is possible
- Add low $t$ coulomb peak to the fit
Elastic scattering experimental data

Preliminary Results

- 30.54 GeV pp
- pp Fit at 30.54 GeV

- 44.64 GeV pp
- pp Fit at 44.64 GeV

- 52.8 GeV pp
- pp Fit at 52.8 GeV

- 62.5 GeV pp
- pp Fit at 62.5 GeV
Single diffractive Scattering

- Incident proton interacts with a target nucleon
- Exits with reduced energy ($M_x$), and an angular kick
- Two regions of $M_x$: baryon resonances at low mass, triple regge at higher mass

\[
\begin{align*}
A & \quad t \\
\downarrow & \downarrow \\
p & \downarrow & a \\
\downarrow & \downarrow & \uparrow \\
b & \quad \xi P & p' \\
\downarrow & \downarrow & \uparrow \\
\downarrow & \downarrow & \uparrow \\
& \quad X & \\
\end{align*}
\]
Cross Sections for "Diffractive" $p + p \rightarrow p + X$ from 100 to 400 GeV (PRL, V34, 1975)

Diffraction Dissociation of High-Energy Protons in p-d Interactions (PRL, V12, 1975)
Triple Regge exchange region

- Fitting over 6000 data points, over 8 experiments!
- More than 70% of the cross section exists at large $M_x$

Preliminary Results

J. Molson, R. Appleby, R. Barlow, A. Toader, M. Serluca

ColUSM #10 Status of loss maps simulations with Merlin 27/28
Conclusions

- We are developing the code Merlin to operate with proton machines for collimations simulations.
- Can now generate loss maps for the LHC!
- When running in sixtrack like scattering mode, similar loss maps are generated for 4TeV 2012 running.
- Scattering fits almost complete, next is to work on p-A interactions - p-Pb run data from the LHC will help.
- New material classes almost complete, will soon be able to simulate new collimator designs.
Aperture configuration

1. Load aperture information file
2. Save data into a vector
3. Extract all AcceleratorModel elements
4. Loop over elements
   - If element is a collimator or has zero length - skip
   - Create a new aperture vector for this element
5. Check if the position of the aperture vector is greater than the element position
6. If we are at the start of the aperture list, extract the aperture from the last entry in the vector and add this to the element specific vector
7. Iterate over the aperture vector whilst we are still inside the component
8. Add one aperture entry past the current AcceleratorComponent
9. Create interpolated RectEllipse aperture and attach to element
10. Create interpolated circular aperture and attach to element
11. Check if each sub-entry is constant along the element length.
12. Check if all entries within each entry are the same.
13. Create RectEllipse aperture and attach to element
14. Create circular aperture and attach to element
15. Move to next element until at the end of the Accelerator
Collimation Process

- Start collimation process
  - Is the current component a collimator?
    - Yes
      - Enable collimation at the end of collimators
        - The cross sections are scaled with energy and other parameters for the jaw material are calculated
        - Check if each particle is within the collimator jaw aperture
          - Outside "beampipe" inside collimator jaw material
            - Inside collimator jaw: Start main scattering routine for the bunch particle type
          - Inside "beampipe" outside collimator jaw material
            - Simulate scattering physics
    - No
      - Enable Collimation at start and middle of element
        - Check if each particle is within the aperture for the element at each point
          - Inside beampipe
            - Particle survives till the end of the collimator
            - Keep particle for further tracking
          - Outside beampipe
            - Particle does not survive till the end of the collimator
            - Remove particle and record location
Collimation Process

1. For non-collimators, create a copy of the bunch at the entry of each element.
2. Log the position in the bunch of any lost particles.
3. Copy the lost particles from the initial conditions bunch copy into the new bunch.
4. Create a new ParticleBunch.
5. While there are still particles remaining and we are inside the element, check aperture.
6. Create a new particle tracker and assign it the new bunch.
7. Add lost particles to the lost particle list as for a collimator.
8. Track a step ds.
9. If at end of element, add any remaining particles to the last bin.
10. Call DoOutput() function and clean up.
Cross sections

Beam momentum (GeV/c) vs. σ (mb)

σ total pp
σ pp elastic
pp total σ
p̅p total σ

J. Molson, R. Appleby, R. Barlow, A. Toader, M. Serluca
ColUSM #10 Status of loss maps simulations with Merlin 33/28