Dire tutorial for Pythia 8 authors

Desired outcomes

- run Dire at least once
- illustrate basic usage
- find technical and physics differences to Pythia simple shower

Since 8.300, Dire is now an embedded part of Pythia.
Dire replaces Pythia's showers by a model that

- has splitting variables that do not distinguish between dipole ends $\rightarrow$ symmetry
- uses (Catani-Seymour) splitting kernels that distinguish between dipole ends $\rightarrow$ deliberate identification of collinear directions
- includes higher-order corrections to kernels

The philosophy behind Dire is

- keep analytic control & make code easily extendable
- do not prevent future use relying on $\overline{\mathrm{MS}}$ renormalization or DGLAP evolution
- supply all ingredients for merging of any (B)SM emission, and beyond NLO

Tutorial: Part 1 of 3: Basic usage

Unfortunately, while preparing the tutorial, a bug was found that makes Dire usage problematic. To fix, you need to edit `include/Pythia8/Dire.h`, changing the function names

```
void endEvent
void beginEvent
```

to

```
void onEndEvent
void onBeginEvent
```

and then recompile Pythia :(

Tutorial: Part 1 of 3: Basic usage

- compile `main300`
- run `./main300 --input main300.cmnd`
  This will produce $e^+e^- \rightarrow$ jets events at LEP.
- Dire allows for weighted showers, to e.g. treat splitting kernel for which overestimates are too low. For final state showers with final-state recoiler, this can (only) happen due to quark mass or $\mathcal{O}(\alpha_s^2)$ corrections
- thus, look at the weight spectrum printed to `weights.dat`
  you could e.g. invoke `gnuplot` via

```
gnuplot -e "set xrange [-5:5]; plot 'weights.dat' u 1:2 w l; pause -1"
```

Tutorial: Part 1 of 3: Basic usage

- In the presence of initial partons, Dire may also produce event weights due to negative PDFs or splitting kernels that turn negative. To test this, it is convenient to switch to a DIS setup, by changing section $3)$ in `main300.cmnd` to

```
# Set hard process.
WeakBosonExchange:ff2ff(t:gmZ) = on
Beams:idA                  = -11
Beams:idB                  = 2212
Beams:frameType            = 2
Beams:eA                   = 27.5
Beams:eB                   = 920.0
PhaseSpace:Q2min           = 1000.0
# Q2 as factorization scale.
SigmaProcess:factorScale2 = 6
SigmaProcess:renormScale2 = 6
```

- run `./main300 --input main300.cmnd`
- look at the weight spectrum again, e.g. with
  `gnuplot -e "set xrange [-5:5]; plot 'weights.dat' u 1:2 w l; pause -1"`
- You should observe a larger spread of event weights. However, the unitarity of the shower guarantees that the average shower weight is still unity.

Tutorial: Part 2 of 3

One main point about Dire is the inclusion of higher-order splitting kernels.

Q: Why bother? **A:** Mainly for matching N^{k}LO calculations.

Example worry: NLO calcuations require NLO PDFs. Initial-state showers should unfold PDF evolution, starting from the PDFs of the hard process $\rightarrow$ require NLO DGLAP kernels.

But is the effect relevant enough that we should worry? $\Rightarrow$ Check!

- Keep the DIS setup of `main300.cmnd`. To avoid having to code the event analysis, you can use the main program `main300plots.cc` from the repository.
- Switch on NLO DGLAP kernels by using the settings

  ```
  DireTimes:kernelOrder = 3
  DireSpace:kernelOrder = 3
  ```

- What's the effect on the pion multiplicity as function of z (as a stand-in for the pion fragmentation function)? `main300plot` will print the histogram table `npi.dat` for you.
- No visible impact on final state, so maybe not something to worry about for the bulk?

Tutorial: Part 2 of 3

Suggested procedure for comparison:

```
#run with default, then
cp npi.dat npi-ko1.dat
#run with new kernel order 3, then
cp npi.dat npi-ko3.dat
# then enter gnuplot and do
plot "< paste npi-ko1.dat npi-ko3.dat" u 1:(($2-$4)/($2+$4)) w l t "default vs. nloglap"
```

Tutorial: Part 3 of 3

However, note that NLO DGLAP cannot be used to regularize NNLO calculations, because it can't act as differential subtraction of IR singularities. The full $\mathcal{O}(\alpha_s^2)$ singularity structure is much more complex, e.g. includes spin- and color-correlations, new phasespace regions to fill, recoil corrections, and all kinds of interference effects.

Double-soft parts of singularity structure are included in Dire – so what's the impact of this compared to the primitive leadin-order shower

- Return to the LEP setup of `main300.cmnd`
- Switch on double-soft kernels by using the settings

```
DireTimes:kernelOrder = 4
DireSpace:kernelOrder = 4
```

- What's the effect on the Durham jet separation, compared to the default result? `main300plot` will print the histogram tables `y23.dat` and `y34.dat` for you.
- Again, overall moderate impact on baseline. Note however that the shower uncertainties will become smaller – and that without playing any tricks with "compensation terms".

Thanks for taking the time!