How to run JETSCAPE to generate the files for 4 $\hat{p}_T$ bins

Goal: Generate 4 different copies of "`jescape_user.xml`" with the $\hat{p}_T$-bin contained in their name.
Reason: These 4 different copies are needed in order to ensure to that ***distinct*** "`cross_section.dat`" and "`test_out.dat`" files are created.

Step 1: Redownload and reinitialize the latest docker container

Outside the docker container

```
mkdir ~/MATTER_LBT_results
```

On Mac OS:

```
docker run -it -v ~/MATTER_LBT_results:/home/jetscape-user/JETSCAPE/MATTER_LBT_results -p 8888:8888 gvujan/jetscape-school:latest
```

On Linux:

```
docker pull gvujan/jetscape-school:latest

docker run -it -v ~/MATTER_LBT_results:/home/jetscape-user/JETSCAPE/MATTER_LBT_results -p 8888:8888 --user $(id -u):$(id -g) gvujan/jetscape-school:latest
```

Steps 2 through 8 are to be done inside the docker container.

Step 2: Open "`jescape_user.xml`" and ensure that you have the following two lines

```
<outputCrossSectionFile>cross_section</outputCrossSectionFile>
<outputFilename>test_out</outputFilename>
```

Step 3: create 4 copies of the "`jescape_user.xml`"

```
cp jetscape_user.xml jetscape_user_100_125.xml
cp jetscape_user.xml jetscape_user_125_150.xml
cp jetscape_user.xml jetscape_user_150_175.xml
cp jetscape_user.xml jetscaep_user_175_200.xml
```

In these "`jetscape_user.xml`" files we will be editing two new arguments:

```
<outputCrossSectionFile>cross_section</outputCrossSectionFile>
<outputFilename>test_out</outputFilename>
```

The role of these arguments is that they allow to change the default output file names for the entire parton evolution profile stored in "`test_out.dat`" as well as the cross-section file (containing the cross section for a given $\hat{p}_T$ bin) stored in "`cross_section.dat`". To ensure that we have 4 different files containing the different $\hat{p}_T$ bins ranging from 100—200 GeV, it is easiest to change the content as is now outlined.

Step 4: Open "`jetscape_user_100_125.xml`" and change the arguments of `<outputCrossSectionFile>` and `<outputFilename>` as specified below

```
<outputCrossSectionFile>cross_section_100_125</outputCrossSectionFile>
<outputFilename>test_out_100_125</outputFilename>
```

Also, change the arguments `<PythiaGun>` to

```
<pTHatMin>100</pTHatMin>
<pTHatMax>125</pTHatMax>
```

Repeat 3 more times, by supplying the appropriate values for

```
jetscape_user_125_150.xml
jetscape_user_150_175.xml
jetscape_user_175_200.xml
```

Step 5: Once done, simply run from the build directory

```
./runJetscape ../config/jetscape_user_100_125.xml
./runJetscape ../config/jetscape_user_125_150.xml
./runJetscape ../config/jetscape_user_150_175.xml
./runJetscape ../config/jetscaep_user_175_200.xml
```

This should generate

```
test_out_100_125.dat  cross_section_100_125.dat
test_out_125_150.dat  cross_section_125_150.dat
test_out_150_175.dat  cross_section_150_175.dat
test_out_175_200.dat  cross_section_175_200.dat
```

Step 6: Once the above files are generated do:

```
cp test_out_100_125.dat cross_section_100_125.dat ~/JETSCAPE/MATTER_LBT_results
cp test_out_125_150.dat cross_section_125_150.dat ~/JETSCAPE/MATTER_LBT_results
cp test_out_150_175.dat cross_section_150_175.dat ~/JETSCAPE/MATTER_LBT_results
cp test_out_175_200.dat cross_section_175_200.dat ~/JETSCAPE/MATTER_LBT_results
```

Step 7: Run "./FinalStateHadrons" as follows and copy the results:

```
./FinalStateHadrons test_out_100_125.dat final_hadrons_100_125.dat
./FinalStateHadrons test_out_125_150.dat final_hadrons_125_150.dat
./FinalStateHadrons test_out_150_175.dat final_hadrons_150_175.dat
./FinalStateHadrons test_out_175_200.dat final_hadrons_175_200.dat
```

And then copy the results over

```
cp final_hadrons_100_125.dat ~/JETSCAPE/MATTER_LBT_results
cp final_hadrons_125_150.dat ~/JETSCAPE/MATTER_LBT_results
cp final_hadrons_150_175.dat ~/JETSCAPE/MATTER_LBT_results
cp final_hadrons_175_200.dat ~/JETSCAPE/MATTER_LBT_results
```

Step 8: Once done, you can start reading the C++ code "`analysis_script.cxx`" in the folder

```
~/JETSCAPE/cross_section_example
```

to familiarize yourself with the code which will be used during the hands-on session on Monday.

Overview for the planned exercise on Monday

In theoretical calculations, the calculation the nuclear modification factor R_{AA} can be done via:

$$R_{AA} = \frac{\left(\frac{1}{\sigma_{inel}} \frac{\Delta\sigma_{AA}}{2\pi p_T \Delta p_T \Delta\eta}\right)}{\left(\frac{1}{\sigma_{inel}} \frac{\Delta\sigma_{pp}}{2\pi p_T \Delta p_T \Delta\eta}\right)}.$$

The most difficult quantity to compute in order to obtain R_{AA} is the numerator. In a fully realistic calculation, we would need many hydrodynamical simulations. For each fluid simulation, many jet simulations are propagated. To illustrate such a calculation given the limited time, we will perform a full jet simulation composed of Pythia initial partons, followed by MATTER and LBT to simulate the parton-QGP fluid interaction on top of one QGP fluid simulation; the latter being obtained from a single fluctuating TRENTO+Free-streaming+Hydro event. The cross section that we will be calculating is going to be obtained through:

$$\frac{1}{\sigma_{inel}} \frac{\Delta\sigma_{AA}}{2\pi p_T \Delta p_T \Delta\eta} = \sum_{k=1}^{4} \frac{\Delta N(\hat{p}_{T;k})}{2\pi p_T \Delta p_T \Delta\eta} \frac{\hat{\sigma}(\hat{p}_{T;k})}{\sigma_{inel}}.$$

The total inelastic pp cross section $\sigma_{inel} = 70$ mb at $\sqrt{s_{NN}} = 5.02$ TeV. Including σ_{inel} in the calculation is optional but makes the numerator of R_{AA} have the same units as pion p_T spectra, for instance. In a tree-level $2 \to 2$ hard scattering, the incoming and outgoing partons are connected via a virtual propagator. That propagator can take many values of 4-momentum giving many possible outgoing particle momenta. The possible values of momenta for outgoing particles are sampled by allowing the propagator to take different transverse momenta ($\hat{p}_T$). Jet energy loss calculations are performed by specifying different ranges, or $\hat{p}_T$-bins, which we have just done in the above exercise using 4 $\hat{p}_T$-bins. These 4 $\hat{p}_T$-bin are labeled by index k, hence the sum spanning values from 1 to 4. The associated cross-section of the $2 \to 2$ process is $\hat{\sigma}(\hat{p}_{T;k})$.

The calculation will be broken down in three steps. We build up ΔN by putting all pions inside each $\hat{p}_T$-bin (labeled by index k in "`analysis_script.cxx`") into different p_T bins (labelled by index j in "`analysis_script.cxx`"). Then, the standard deviation associated with each (k,j) combination will be calculated before the k-labelled bins are combined (i.e. added) together in the final step giving us the final spectrum $\frac{1}{\sigma_{inel}} \frac{\Delta\sigma_{AA}}{2\pi p_T \Delta p_T \Delta\eta}$, which can be compared against experimental data. Details about each step are specified below and each step will also be indicated in the provided C++ code named "`analysis_script.cxx`", which can be found inside the docker container under:

```
~/JETSCAPE/cross_section_example
```

Steps to be taken on Monday inside the docker container

Step 1: For each $\hat{p}_T$-bin whose range as specified in the `jetscape_user.xml` files, if a pion is found in final_hadrons.dat, add a +1 counter to the p_T range specified pTMin, pTMax, and pTBinW (see "`analysis_script.cxx`" for definition of pTBinW). This will fill out the array dNdpTCount_hard_ind[NpTHatBin][NpTBin].

(Note: If we had <recoil_on> 1 </recoil_on> instead of <recoil_on> 0 </recoil_on> in MATTER and/or MARTINI through the in the `jetscape_user.xml` files, we would also have to fill dNdpTCount_soft_ind[NpTHatBin][NpTBin], for now leave it empty.)

Step 2: For each $\hat{p}_T$ (index k), once all pions have been binned according their p_T (index j), calculate the standard deviation on their cross-section using gaussian propagation of uncertainty assuming uncorrelated uncertainties. Assume that the variance associated with assigning a particle to a p_T-bin is $\sigma^2 = \frac{1}{\Delta N}$. The standard deviation associated with each $\hat{\sigma}(\hat{p}_T)$ is found in "`cross_section.dat`" file. Put your answer in the array CS_Err_hard_ind[k][j].

(Note: If we had <recoil_on> 1 </recoil_on> instead of <recoil_on> 0 </recoil_on> in MATTER and/or MARTINI through the in the `jetscape_user.xml` files, we would also have to fill CS_Err_soft_ind[k][j], for now leave it empty.)

Step 3: Combine (i.e. add) the different bins, weighed by $\hat{\sigma}(\hat{p}_T)$, and assume fully uncorrelated Gaussian uncertainty propagation between $\hat{p}_T$ bins. You can choose to either fill the temporary array CS_hard_ind[j] or directly fill TotalCS[j].

(Note: If we had <recoil_on> 1 </recoil_on> instead of <recoil_on> 0 </recoil_on> in MATTER and/or MARTINI through the in the `jetscape_user.xml` files, we could also have to fill the temporary array CS_soft_ind[j], for now leave it empty.)