

JETSCAPE Hydrodynamics Session

Goals

- Understand how to run JETSCAPE with a few default Fluid dynamics modules and set/change their relevant parameters, such as transport coefficients
- With a realistic hydrodynamic module, students will be able to output hydrodynamic evolution profile and analyze the evolution of temperature and the development of flow velocity with various settings
- Simulate event-by-event bulk dynamics for Au+Au @ 200 GeV and Pb+Pb @ 5020 GeV with a realistic hydrodynamics code, such as MUSIC.

Physics Background

The JETSCAPE framework employs the Trento model to generate event-by-event initial energy density profile. The energy density profile is then passed to the hydrodynamics module (MUSIC), which will evolve the collision system from a hot QGP phase to the hadron gas phase. Optionally, pre-equilibrium dynamics modeled by free-streaming can be included between the Trento and hydrodynamics. In the dilute hadronic phase, fluid cells will convert to individual hadrons. This process is denoted as the particlization. The JETSCAPE framework uses iSS to perform particlization. The produced hadrons can be fed to a hadronic transport model (SMASH), which accounts for scattering processes among hadrons and decays of excited resonance states.

Setup a docker container

If you plan to do exercises on a remote computer with `ssh`, please use the following command to log in to your remote machine,

```
ssh -L 8888:127.0.0.1:8888 user@server
```

The port information is essential to properly setup jupyter notebook.

Before we begin our session, please make sure all the code packages are already in the correct place on your computer. You should have a `jetscape-docker` folder under your home directory. Try to list the folder inside `jetscape-docker` with the following command,

```
ls ~/jetscape-docker
```

You need to make sure the following folders are present,

- JETSCAPE
- SummerSchool2020

```
chunshen@Chuns-MacBook-Pro ~/jetscape-docker$ ls ~/jetscape-docker
JETSCAPE      SummerSchool12020
chunshen@Chuns-MacBook-Pro ~/jetscape-docker$
```

In this session, we need to launch a docker container that supports the jupyter notebook. Please use the following command:

```
macOS: docker run -it -p 8888:8888 -v ~/jetscape-docker:/home/jetscape-user --name myJSHydroSession jetscape/base:v1.4
```

Linux:

```
docker run -it -p 8888:8888 -v ~/jetscape-docker:/home/jetscape-user --name myJSHydroSession --user $(id -u):$(id -g) jetscape/base:v1.4
```

- `--rm` This option will delete the current docker container at the exit. (If you want to delete the container, you can add this option.)
- `-p 8888:8888` This option creates a port for your web browser outside the docker container to load a jupyter notebook. All the python packages are in the docker container.

Under Linux, if you encounter an error about `permission denied`, you can use `sudo` in front of the docker run command.

Build JETSCAPE with MUSIC and iSS

We will do all of our exercises in the `JETSCAPE/build` directory. Please make sure all the external code packages (MUSIC and iSS) have been downloaded. You can check this by the following commands,

```
cd ~/JETSCAPE/external_packages
ls
```

Please check the folder `music` and `iss` are present.

```
jetscape-user@3871e6d29f2c:~$ cd ~/JETSCAPE/external_packages
jetscape-user@3871e6d29f2c:~/JETSCAPE/external_packages$ ls
CMakeLists.txt  cornelius.h  get_freestream-milne.sh  get_smash.sh  gzstream.h  sigslot.h  trento
README.md       fjcore.cc    get_iss.sh             googletest    hydro_from_external_file  smash
clvisc_wrapper  fjcore.hh    get_lbtTab.sh         gtl           iss          tinyxml2.cc
cornelius.cpp   get_clvisc.sh  get_music.sh         gzstream.cc   music       tinyxml2.h
jetscape-user@3871e6d29f2c:~/JETSCAPE/external_packages$
```

If not, please run the following commands,

```
./get_music.sh
./get_iss.sh
```

When you enter the docker container, type the following commands to setup the working directory,

```
cd ~/JETSCAPE
mkdir -p build
cd build
cmake .. -DUSE_MUSIC=ON -DUSE_ISS=ON
make -j4
cp -r ../../SummerSchool2020/hydro_session .
```

The last command copies the hand-on materials of this session to the working folder.

```
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ cp -r ../../SummerSchool2020/hydro_session .
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ ls
CMakeCache.txt  FinalStateHadrans  cmake_install.cmake  external_packages  iss_tables  readerTest
CMakeFiles      FinalStatePartons  data_table           hydro_session      lib         runJetscape
EOS             Makefile           examples             iss_parameters.dat  music_input  src
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$
```



1. A Test Run for JETSCAPE with MUSIC

To perform a test run for JETSCAPE with MUSIC in our working directory (`~/JETSCAPE/build`),

```
cd ~/JETSCAPE/build
./runJetscape hydro_session/jetscape_user_MUSICTestRun.xml
```

```
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ cd ~/JETSCAPE/build
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ ./runJetscape hydro_session/jetscape_user_MUSICTestRun.xml
```

```

1 <?xml version="1.0"?>
2
3 <jetscape>
4 <!-- General settings -->
5 <nEvents> 1 </nEvents>
6
7 <!-- fix the random seed -->
8 <Random>
9 <seed>23</seed>
10 </Random>
11
12 <!-- Inital State Module -->
13 <IS>
14 <!-- set boost invariant simulation -->
15 <grid_max_z> 0.0 </grid_max_z>
16
17 <!-- Initial condition -->
18 <Trento> </Trento> ← Initial condition
19 </IS>
20
21 <!--Preequilibrium Dynamics Module -->
22 <Preequilibrium>
23 <NullPreDynamics> </NullPreDynamics> ← Hydrodynamics
24 </Preequilibrium> module
25
26 <!-- Hydro Module -->
27 <Hydro>
28 <MUSIC>
29 <freezeout_temperature>0.150</freezeout_temperature>
30 </MUSIC>
31 </Hydro>
32 </jetscape>
33

```

Visualization with Jupyter Notebook

Launch jupyter notebook inside the docker contain with the following command in our working directory (`~/JETSCAPE/build`),

```

cd ~/JETSCAPE/build
jupyter notebook --ip 0.0.0.0 --no-browser > notebook.log 2>&1 &
cat notebook.log

```

Once the jupyter notebook is running in the background, the user can click on the link by holding the `ctrl` key. The link is displayed at the second to the last line and begins with `http://127.0.0.1:8888/?token=...` After the click, your web browser should be launched in the current directory. If your terminal does not recognize html addresses, you can open your browser and enter the following address, `http://127.0.0.1:8888/?token=...`

```

jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ cd ~/JETSCAPE/build
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ jupyter notebook --ip 0.0.0.0 --no-browser > notebook.log 2>&1 &
[1] 1670
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ cat notebook.log
[I 21:25:29.153 NotebookApp] Writing notebook server cookie secret to /home/jetscape-user/.local/share/jupyter/runtime/notebook_cookie_secret
[I 21:25:30.789 NotebookApp] Serving notebooks from local directory: /home/jetscape-user/JETSCAPE/build
[I 21:25:30.790 NotebookApp] The Jupyter Notebook is running at:
[I 21:25:30.790 NotebookApp] http://3871e6d29f2c:8888/?token=053f423c59290a1501a106c7911906135ad9508172776ce9
[I 21:25:30.791 NotebookApp] or http://127.0.0.1:8888/?token=053f423c59290a1501a106c7911906135ad9508172776ce9
[I 21:25:30.792 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:25:30.818 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/jetscape-user/.local/share/jupyter/runtime/nbserver-1670-open.html
Or copy and paste one of these URLs:
http://3871e6d29f2c:8888/?token=053f423c59290a1501a106c7911906135ad9508172776ce9
or http://127.0.0.1:8888/?token=053f423c59290a1501a106c7911906135ad9508172776ce9
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$

```

Copy and paste this address to your browser

Files Running Clusters

Select items to perform actions on them. Upload New ↕

Name	Last Modified	File size
0 /		
CMakeFiles	19 minutes ago	
data_table	33 minutes ago	
EOS	33 minutes ago	
examples	33 minutes ago	
external_packages	33 minutes ago	
hydro_session	15 minutes ago	
ISS_tables	33 minutes ago	
lib	27 minutes ago	
src	33 minutes ago	
cmake_install.cmake	33 minutes ago	5.04 kB
CMakeCache.txt	33 minutes ago	35.8 kB
FinalStateHadrons	20 minutes ago	567 kB
FinalStatePartons	20 minutes ago	607 kB
iSS_parameters.dat	40 minutes ago	10.8 kB
Makefile	33 minutes ago	32.2 kB
music_input	an hour ago	9.46 kB
notebook.log	seconds ago	1.16 kB
readerTest	20 minutes ago	835 kB
runJetscape	19 minutes ago	674 kB

click on this folder

1. Plot averaged temperature and flow velocity evolution

In your browser, we first go into the `hydro_session` folder (in your browser). We can open the notebook `hydro_evo_TestRun.ipynb` by click on it inside the browser. Once the notebook is opened, the user can execute every cell in this notebook one-by-one. Press **shift+enter** to execute the cell block in the notebook.

Files Running Clusters

Select items to perform actions on them. Upload New ↕

Name	Last Modified	File size
0 / hydro_session		
..	seconds ago	
figs	18 minutes ago	
analyze_particle_spectra_and_vn.ipynb	17 minutes ago	170 kB
hydro_evo-CollisionSystemsComp.ipynb	17 minutes ago	365 kB
hydro_evo-TestRun.ipynb	17 minutes ago	300 kB
hydro_evo-ViscosityComp.ipynb	17 minutes ago	344 kB
hydro_movie-TempDepVisc.ipynb	17 minutes ago	415 kB
hydro_movie-TestRun.ipynb	17 minutes ago	367 kB
analyze_particle_spectra_and_vn.py	17 minutes ago	3.91 kB

First click and start this notebook ...

2. Animation of averaged temperature and flow velocity evolution

Similar to the previous exercise, this time we will open the second notebook under `hydro_session` folder, `hydro_movie_TestRun.ipynb`. After it is opened inside your browser, you can execute the code cells one by one to generate 2D color contour plots as well as animation for the temperature and fluid velocity evolution.

Files Running Clusters

Select items to perform actions on them. Upload New ↻

0	hydro_session	Name	Last Modified	File size
0	..		seconds ago	
0	figs		21 minutes ago	
0	analyze_particle_spectra_and_vn.ipynb		21 minutes ago	170 kB
0	hydro_evo-CollisionSystemsComp.ipynb		21 minutes ago	365 kB
0	hydro_evo-TestRun.ipynb		21 minutes ago	300 kB
0	hydro_evo-ViscosityComp.ipynb		21 minutes ago	344 kB
0	hydro_movie-TempDepVisc.ipynb		21 minutes ago	415 kB
0	hydro_movie-TestRun.ipynb		21 minutes ago	367 kB
0	analyze_particle_spectra_and_vn.py		21 minutes ago	3.91 kB
0	collect_results.sh		21 minutes ago	240 B

← Now click this notebook

No Jupyter Notebook?

If you can not use jupyter notebook, there are python scripts in the hydro_session folder to generate the same plots and animation. Users can run these scripts as follows,

```
cd ~/JETSCAPE/build/hydro_session
python3 hydro_evo-TestRun.py
python3 hydro_movie-TestRun.py
```

```
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ cd ~/JETSCAPE/build/hydro_session
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ python3 hydro_evo-TestRun.py
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ python3 hydro_movie-TestRun.py
Read in data completed.
ntau = 68, tau_min = 0.50 fm, tau_max = 7.20 fm, dtau = 0.100 fm
nx = 50, x_min = -15.00 fm, x_max = 14.40 fm, dx = 0.60 fm
ny = 50, y_min = -15.00 fm, y_max = 14.40 fm, dy = 0.60 fm
neta = 1, eta_min = 0.00 fm, eta_max = 0.00 fm, deta = 0.10
Generate 2D contour plot for T x vs y ...
Generate 2D contour plot for T tau vs x ...
Generate animation for T ...
Generate animation for T and flow ...
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ ls
README.md                                animation_temperature.mp4                hydro_movie-TempDepVisc.py
TestRun_Temperature_Contour_TauX.pdf      collect_results.sh                       hydro_movie-TestRun.ipynb
TestRun_Temperature_Contour_XY.pdf        figs                                     hydro_movie-TestRun.py
TestRun_avgT_evo.pdf                     hydro_evo-CollisionSystemsComp.ipynb    jetscape_user_AuAu200.xml
TestRun_avgv_evo.pdf                     hydro_evo-CollisionSystemsComp.py       jetscape_user_MUSICTestRun.xml
TestRun_ecc_evo.pdf                       hydro_evo-TestRun.ipynb                 jetscape_user_MUSIC_and_iSS.xml
TestRun_momentumaniso_evo.pdf            hydro_evo-TestRun.py                    jetscape_user_PbPb5020.xml
analyze_particle_spectra_and_vn.ipynb    hydro_evo-ViscosityComp.ipynb          jetscape_user_TempDepVis.xml
analyze_particle_spectra_and_vn.py       hydro_evo-ViscosityComp.py              jetscape_user_shear.xml
animation_TwithFlow.mp4                  hydro_movie-TempDepVisc.ipynb          jetscape_user_shear_and_bulk.xml
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$
```

← These steps could be slow on your computer



2. Change the collision systems

The users can specify the types of collision systems in JETSCAPE xml file,

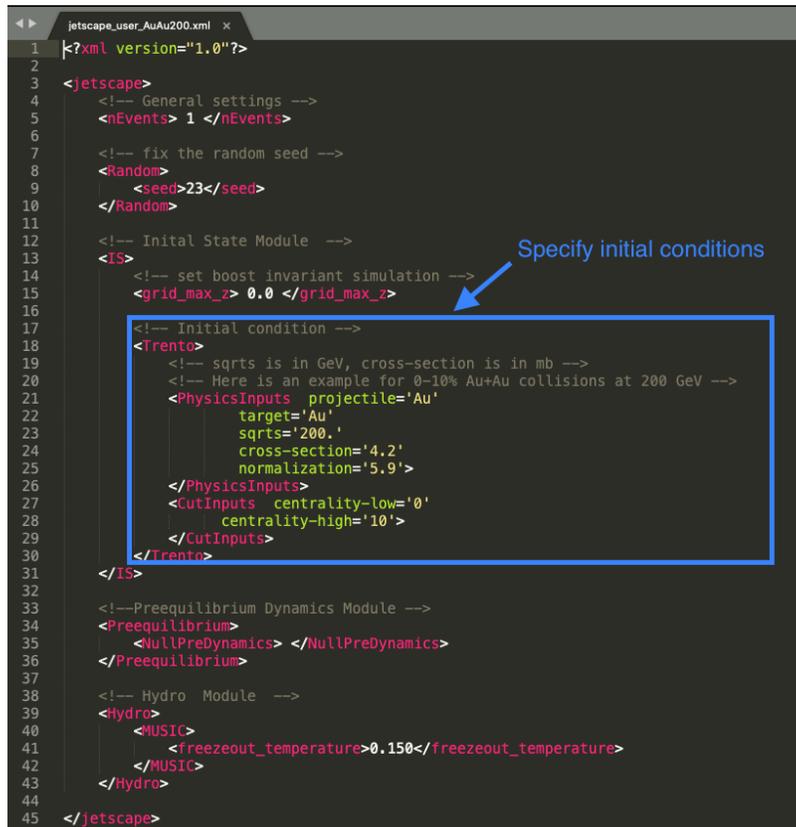
- Collision Energy
- Colliding Nuclei
- Centrality

In the user configuration file hydro_session/jetscape_user_AuAu200.xml, one can modify model parameters for the initial state module to simulate his/her desired collision system. In between <Trento> and <\Trento>, we can specify the type of colliding nucleus, collision energy, and centrality.

[Run 1] Simulate a 0-10% Au+Au collision at 200 GeV in our working directory (~/JETSCAPE/build),

```
cd ~/JETSCAPE/build
./runJetscape hydro_session/jetscape_user_AuAu200.xml
./hydro_session/collect_results.sh Run_AuAu200_C0-10
```

The last command collects all the results into a new folder, `Run_AuAu200_C0-10` .



```
1 <?xml version="1.0"?>
2
3 <jetscape>
4   <!-- General settings -->
5   <nEvents> 1 </nEvents>
6
7   <!-- fix the random seed -->
8   <Random>
9     <seed>23</seed>
10  </Random>
11
12  <!-- Initial State Module -->
13  <IS>
14    <!-- set boost invariant simulation -->
15    <grid_max_z> 0.0 </grid_max_z>
16
17    <!-- Initial condition -->
18    <Trento>
19      <!-- sqrts is in GeV, cross-section is in mb -->
20      <!-- Here is an example for 0-10% Au+Au collisions at 200 GeV -->
21      <PhysicsInputs projectile='Au'
22                    target='Au'
23                    sqrts='200.'
24                    cross-section='4.2'
25                    normalization='5.9' >
26      </PhysicsInputs>
27      <CutInputs centrality-low='0'
28                centrality-high='10' >
29      </CutInputs>
30    </Trento>
31  </IS>
32
33  <!-- Preequilibrium Dynamics Module -->
34  <Preequilibrium>
35    <NullPreDynamics> </NullPreDynamics>
36  </Preequilibrium>
37
38  <!-- Hydro Module -->
39  <Hydro>
40    <MUSIC>
41      <freezeout_temperature>0.150</freezeout_temperature>
42    </MUSIC>
43  </Hydro>
44
45 </jetscape>
```

Here the nucleon-nucleon cross section is specified in the unit (fm²) (1[fm²]= 10[mb])

[Run 2] Simulate a 20-30% Pb+Pb collisions at 5.02 TeV in our working directory (`-/JETSCAPE/build`),

```
./runJetscape hydro_session/jetscape_user_PbPb5020.xml
./hydro_session/collect_results.sh Run_PbPb5020_C20-30
```

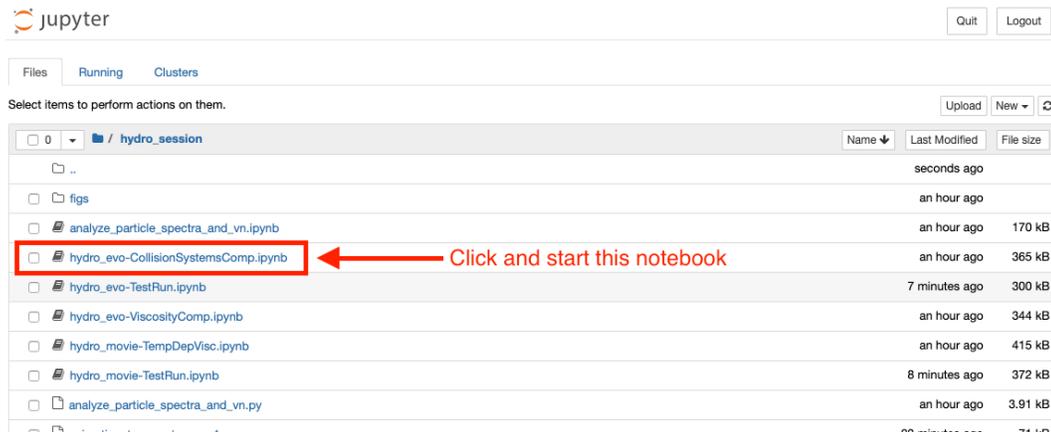
```

1 <?xml version="1.0"?>
2
3 <jetscape>
4 <!-- General settings -->
5 <nEvents> 1 </nEvents>
6
7 <!-- fix the random seed -->
8 <Random>
9 <seed>23</seed>
10 </Random>
11
12 <!-- Initial State Module -->
13 <IS>
14 <!-- set boost invariant simulation -->
15 <grid_max_z> 0.0 </grid_max_z>
16
17 <!-- Initial condition -->
18 <Trento>
19 <!-- sqrts is in GeV, cross-section is in mb -->
20 <!-- Here is an example for 20-30% Pb+Pb collisions at 5.02 TeV -->
21 <PhysicsInputs projectile='Pb'
22 target='Pb'
23 sqrts='5020.'
24 cross-section='6.7'
25 normalization='18.4'>
26 </PhysicsInputs>
27 <CutInputs centrality-low='20'
28 centrality-high='30'>
29 </CutInputs>
30 </Trento>
31 </IS>
32
33 <!-- Preequilibrium Dynamics Module -->
34 <Preequilibrium>
35 <NullPreDynamics> </NullPreDynamics>
36 </Preequilibrium>
37
38 <!-- Hydro Module -->
39 <Hydro>
40 <MUSIC>
41 <freezeout_temperature>0.150</freezeout_temperature>
42 </MUSIC>
43 </Hydro>
44
45 </jetscape>
46

```

Specify initial conditions

Now we can compare the evolution temperature and flow velocity between these two systems. You can visualize the comparison using the jupyter notebook, `hydro_evo-CollisionSystemsComp.ipynb`. Launch this notebook in your browser and run the code cells one-by-one to see the comparisons (same procedure as in the previous exercise). Alternatively, you can directly run `hydro_evo-CollisionSystemsComp.py` to generate the same plots.



The results plots can be found in the `hydro_session` folder,

```

jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ python3 hydro_evo-CollisionSystemsComp.py
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ ls
README.md          analyze_particle_spectra_and_vn.py  hydro_movie-TempDepVisc.py
RunCollisionSystemComp_avgv_evo.pdf  animation_TwithFlow.mp4           hydro_movie-TestRun.ipynb
RunCollisionSystemComp_ecc2_evo.pdf   animation_temperature.mp4         hydro_movie-TestRun.py
RunCollisionSystemComp_momentumniso_evo.pdf  collect_results.sh                jetscape_user_AuAu200.xml
RunCollisionSystemsComp_avgT_evo.pdf  figs                               jetscape_user_MUSICTestRun.xml
TestRun_Temperature_Contour_TauX.pdf   hydro_evo-CollisionSystemsComp.ipynb  jetscape_user_MUSIC_and_ISS.xml
TestRun_avgT_evo.pdf                  hydro_evo-CollisionSystemsComp.py   jetscape_user_PbPb5020.xml
TestRun_avgv_evo.pdf                  hydro_evo-TestRun.ipynb           jetscape_user_TempDepVis.xml
TestRun_ecc_evo.pdf                   hydro_evo-TestRun.py              jetscape_user_shear.xml
TestRun_momentumniso_evo.pdf          hydro_evo-ViscosityComp.ipynb     jetscape_user_shear_and_bulk.xml
analyze_particle_spectra_and_vn.ipynb  hydro_movie-TempDepVisc.ipynb
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$

```



3. Study the effects of viscosity in hydrodynamic evolution

- Specific shear viscosity η / s
- Specific bulk viscosity ζ / s

Using a realistic hydrodynamic module inside the JETSCAPE, the users have freedom to change a few physical parameters for the hydrodynamic simulations. The most interesting ones are the specific shear and bulk viscosity. All the relevant parameters are under the block `<hydro>` `<MUSIC>`.

The user can set a constant specific shear viscosity by changing the value for the parameter `<shear_viscosity_eta_over_s>`. A physical η/s needs to be a positive value. We recommend the users to try any real value between 0 and 0.3 in the exercises.

Moreover, the users can include a non-zero bulk viscosity in the hydrodynamic simulations. Because the QCD bulk viscosity is related to the breaking of conformal symmetry, we expect the specific bulk viscosity to depend on temperature, $(\zeta / s)(T)$.

Users can set the parameter `<temperature_dependent_bulk_viscosity>` to 1, in the xml file to include a temperature dependent $(\zeta / s)(T)$. Setting `<temperature_dependent_bulk_viscosity>` to 0 will set $\zeta / s = 0$.

Users can run the JETSCAPE with two example config files,

```
cd ~/JETSCAPE/build
./runJetscape hydro_session/jetscape_user_shear.xml
./hydro_session/collect_results.sh Run_shear_only
./runJetscape hydro_session/jetscape_user_shear_and_bulk.xml
./hydro_session/collect_results.sh Run_shear_and_bulk
```

```
1 <?xml version="1.0"?>
2
3 <jetscape>
4 <!-- General settings -->
5 <nEvents> 1 </nEvents>
6
7 <Random>
8 <seed>23</seed>
9 </Random>
10
11 <!-- Inital State Module -->
12 <IS>
13 <!-- set boost invariant simulation -->
14 <grid_max_z> 0.0 </grid_max_z>
15
16 <!-- Initial condition -->
17 <Trento>
18 <!-- sqrts is in GeV, cross-section is in mb -->
19 <!-- Here is an example for 20-30% Pb+Pb collisions at 2.76 TeV -->
20 <PhysicsInputs projectile='Pb'
21 target='Pb'
22 sqrts='2760.'
23 cross-section='6.4'
24 normalization='13.9'>
25 </PhysicsInputs>
26 <CutInputs centrality-low='20'
27 centrality-high='30'>
28 </CutInputs>
29 </Trento>
30 </IS>
31
32 <!-- Preequilibrium Dynamics Module -->
33 <Preequilibrium>
34 <NullPreDynamics> </NullPreDynamics>
35 </Preequilibrium>
36
37 <!-- Hydro Module -->
38 <Hydro>
39 <MUSIC>
40 <name>MUSIC</name>
41 <shear_viscosity_eta_over_s>0.15</shear_viscosity_eta_over_s>
42 <temperature_dependent_bulk_viscosity>1</temperature_dependent_bulk_viscosity>
43 <freezeout_temperature>0.150</freezeout_temperature>
44 </MUSIC>
45 </Hydro>
46
47 </jetscape>
48
```

Comparison plots can be made using the jupyter notebook `hydro_evo-ViscosityComp.ipynb` or `hydro_evo-ViscosityComp.py`.

Files Running Clusters

Select items to perform actions on them. Upload New ↻

	Name	Last Modified	File size
0	/ hydro_session		
	..	seconds ago	
	figs	an hour ago	
	analyze_particle_spectra_and_vn.ipynb	an hour ago	170 kB
	hydro_evo-CollisionSystemsComp.ipynb	16 minutes ago	358 kB
	hydro_evo-TestRun.ipynb	36 minutes ago	300 kB
	hydro_evo-ViscosityComp.ipynb	an hour ago	344 kB
	hydro_movie-TempDepVisc.ipynb	an hour ago	415 kB
	hydro_movie-TestRun.ipynb	37 minutes ago	372 kB
	analyze_particle_spectra_and_vn.py	an hour ago	3.91 kB

← Click and start this notebook

The resulting plots can be found under the `hydro_session` folder.

```
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ python3 hydro_evo-ViscosityComp.py
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ ls
README.md                               TestRun_ecc_evo.pdf                    hydro_evo-ViscosityComp.py
RunCollisionSystemComp_avgv_evo.pdf      TestRun_momentumaniso_evo.pdf          hydro_movie-TempDepVisc.ipynb
RunCollisionSystemComp_ecc2_evo.pdf      analyze_particle_spectra_and_vn.ipynb  hydro_movie-TempDepVisc.py
RunCollisionSystemComp_momentumaniso_evo.pdf  analyze_particle_spectra_and_vn.py    hydro_movie-TestRun.ipynb
RunCollisionSystemsComp_avgT_evo.pdf      animation_TwithFlow.mp4                hydro_movie-TestRun.py
RunViscosityComp_avgT_evo.pdf            collect_results.sh                      jetscape_user_AuAu200.xml
RunViscosityComp_avgv_evo.pdf            figs                                     jetscape_user_MUSICTestRun.xml
RunViscosityComp_ecc2_evo.pdf            hydro_evo-CollisionSystemsComp.ipynb   jetscape_user_MUSIC_and_iSS.xml
RunViscosityComp_momentumaniso_evo.pdf    hydro_evo-CollisionSystemsComp.py      jetscape_user_PbPb5020.xml
TestRun_Temperature_Contour_TauX.pdf      hydro_evo-TestRun.ipynb                jetscape_user_TempDepVis.xml
TestRun_Temperature_Contour_XY.pdf        hydro_evo-TestRun.py                   jetscape_user_shear.xml
TestRun_avgT_evo.pdf                     hydro_evo-ViscosityComp.ipynb          jetscape_user_shear_and_bulk.xml
TestRun_avgv_evo.pdf
```



4. Temperature dependent $(\eta/s)(T)$ and $(\zeta/s)(T)$

One can further try a temperature-dependent $(\eta/s)(T)$ by setting the variable `<T_dependent_Shear_to_S_ratio>` to 3. Once this parameter is set to 3, the previous parameter `<shear_viscosity_eta_over_s>` will be ineffective. The users need to further specify the following four parameters,

1. `<eta_over_s_T_kink_in_GeV>`
2. `<eta_over_s_low_T_slope_in_GeV>`
3. `<eta_over_s_high_T_slope_in_GeV>`
4. `<eta_over_s_at_kink>`

to characterize the temperature dependence of $(\eta/s)(T)$.

$$\frac{\eta}{s} \Big|_{\text{lin}}(T) = a_{\text{low}}(T - T_\eta)\Theta(T_\eta - T) + (\eta/s)_{\text{kink}} + a_{\text{high}}(T - T_\eta)\Theta(T - T_\eta).$$

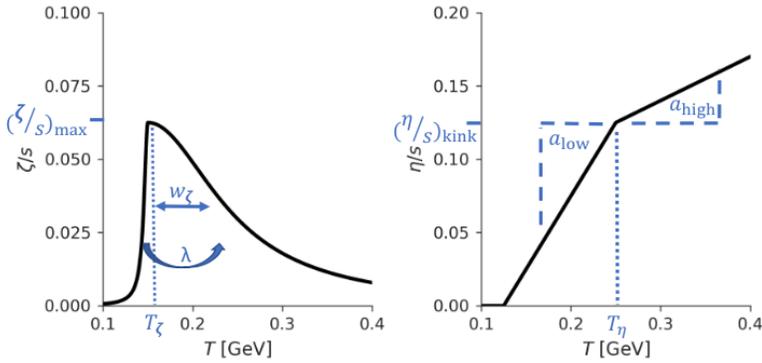
Similar to the case for $(\eta/s)(T)$, the users can set the parameter `<temperature_dependent_bulk_viscosity>` to 3 in the xml file to include a temperature dependent $(\zeta/s)(T)$. With `<temperature_dependent_bulk_viscosity>` set to 3, the users need to further provide four additional parameters to characterize the shape of $(\zeta/s)(T)$. They are as follows,

1. `<zeta_over_s_max>`
2. `<zeta_over_s_T_peak_in_GeV>`
3. `<zeta_over_s_width_in_GeV>`
4. `<zeta_over_s_lambda_asymm>`

The parameterization is

$$\frac{\zeta}{s}(T) = \frac{(\zeta/s)_{\max} \Lambda^2}{\Lambda^2 + (T - T_{\zeta,c})^2},$$

$$\Lambda = w_{\zeta} [1 + \lambda \text{sign}(T - T_{\zeta,c})].$$



Users can play with settings in `hydro_session/jetscape_user_TempDepVis.xml`.

```

jetscape_user_TempDepVis.xml
<!-- Random seed -->
</Random>

<!-- Initial State Module -->
<IS>
  <!-- x range [-grid_max_x, grid_max_x] -->
  <!-- y range [-grid_max_y, grid_max_y] -->
  <!-- longitudinal range [-grid_max_z, grid_max_z] -->
  <!-- in units of [fm] -->
  <grid_max_x> 12 </grid_max_x>
  <grid_max_y> 12 </grid_max_y>
  <grid_max_z> 0.0 </grid_max_z>
  <grid_step_x> 0.2 </grid_step_x>
  <grid_step_y> 0.2 </grid_step_y>
  <grid_step_z> 0.2 </grid_step_z>

  <!-- Initial condition -->
  <Trento>
    <PhysicsInputs projectile='Pb'
                  target='Pb'
                  sqrts='5020'
                  cross-section='6.7'
                  normalization='13.9'>
    </PhysicsInputs>
    <CutInputs centrality-low='20'
              centrality-high='30'>
    </CutInputs>
  </Trento>
</IS>

<!-- Preequilibrium Dynamics Module -->
<Preequilibrium>
  <NullPreDynamics> </NullPreDynamics>
</Preequilibrium>

<!-- Hydro Module -->
<Hydro>
  <MUSIC>
    <name>MUSIC</name>
    <T_dependent_Shear_to_S_ratio>3</T_dependent_Shear_to_S_ratio>
    <eta_over_s_T_kink_in_GeV>0.16</eta_over_s_T_kink_in_GeV>
    <eta_over_s_low_T_slope_in_GeV>0.1</eta_over_s_low_T_slope_in_GeV>
    <eta_over_s_high_T_slope_in_GeV>0.2</eta_over_s_high_T_slope_in_GeV>
    <eta_over_s_at_kink>0.08</eta_over_s_at_kink>
    <temperature_dependent_bulk_viscosity>3</temperature_dependent_bulk_viscosity>
    <zeta_over_s_max>0.1</zeta_over_s_max>
    <zeta_over_s_T_peak_in_GeV>0.16</zeta_over_s_T_peak_in_GeV>
    <zeta_over_s_width_in_GeV>0.01</zeta_over_s_width_in_GeV>
    <zeta_over_s_lambda_asymm>0.0</zeta_over_s_lambda_asymm>
    <Include_second_order_terms>1</Include_second_order_terms>
    <freezeout_temperature>0.150</freezeout_temperature>
  </MUSIC>
</Hydro>
</jetscape>

```

These parameters control the temperature-dependent shear and bulk viscosities

```
cd ~/JETSCAPE/build
./runJetscape hydro_session/jetscape_user_TempDepVis.xml
./hydro_session/collect_results.sh Run_TempDepVisc
```

The associated plotting scripts are `hydro_movie-TempDepVisc.ipynb` and `hydro_movie-TempDepVisc.py`.

The screenshot shows the JupyterLab interface with the 'hydro_session' directory selected. The file list includes:

- ..
- figs
- analyze_particle_spectra_and_vn.ipynb
- hydro_evo-CollisionSystemsComp.ipynb
- hydro_evo-TestRun.ipynb
- hydro_evo-ViscosityComp.ipynb
- hydro_movie-TempDepVisc.ipynb** (highlighted)
- hydro_movie-TestRun.ipynb
- analyze_particle_spectra_and_vn.py

```
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ python3 hydro_movie-TempDepVisc.py
Read in data completed.
ntau = 110, tau_min = 0.50 fm, tau_max = 11.40 fm, dtau = 0.100 fm
nx = 60, x_min = -12.00 fm, x_max = 11.60 fm, dx = 0.40 fm
ny = 60, y_min = -12.00 fm, y_max = 11.60 fm, dy = 0.40 fm
neta = 1, eta_min = 0.00 fm, eta_max = 0.00 fm, deta = 0.10
Generate 2D contour plot for T x vs y ...
Generate 2D contour plot for T tau vs x ...
Generate animation for T ...
Generate animation for T and flow ...
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ ls
README.md                               TestRun_Temperature_Contour_XY.pdf    hydro_evo-TestRun.py
RunCollisionSystemComp_avgv_evo.pdf     TestRun_avgT_evo.pdf                 hydro_evo-ViscosityComp.ipynb
RunCollisionSystemComp_ecc2_evo.pdf     TestRun_avgv_evo.pdf                 hydro_evo-ViscosityComp.py
RunCollisionSystemComp_momentumaniso_evo.pdf TestRun_ecc_evo.pdf                 hydro_movie-TempDepVisc.ipynb
RunCollisionSystemsComp_avgT_evo.pdf    TestRun_momentumaniso_evo.pdf        hydro_movie-TempDepVisc.py
RunTempDepVisc_Temperature_Contour_TauX.pdf analyze_particle_spectra_and_vn.ipynb hydro_movie-TestRun.ipynb
RunTempDepVisc_Temperature_Contour_XY.pdf analyze_particle_spectra_and_vn.py    hydro_movie-TestRun.py
RunTempDepVisc_animation_T.mp4         animation_temperature.mp4           jetscape_user_AuAu200.xml
RunTempDepVisc_animation_TwithFlow.mp4  animation_TwithFlow.mp4             jetscape_user_MUSICTestRun.xml
RunViscosityComp_avgT_evo.pdf           collect_results.sh                   jetscape_user_MUSIC_and_ISS.xml
RunViscosityComp_avgv_evo.pdf           figs                                  jetscape_user_PbPb5020.xml
RunViscosityComp_ecc2_evo.pdf           hydro_evo-CollisionSystemsComp.ipynb jetscape_user_TempDepVis.xml
RunViscosityComp_momentumaniso_evo.pdf  hydro_evo-CollisionSystemsComp.py    jetscape_user_shear.xml
TestRun_Temperature_Contour_TauX.pdf    hydro_evo-TestRun.ipynb             jetscape_user_shear_and_bulk.xml
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$
```

Side notes

In addition to specific shear and bulk viscosity, the users have freedom to change the starting time of hydrodynamics, `<Initial_time_tau_0>` (0.2-1.0 fm), whether to include second order transport coefficients, `<Include_second_order_terms>` (0 or 1), and particlization temperature, `<freezeout_temperature>` (0.13 to 0.16 GeV)

Initial state module, between `<IS>` and `</IS>`, also define the 3D grid that we would like to simulate hydrodynamic evolution. If one set `grid_max_z` to 0, the JETSCAPE framework will perform a (2+1) hydrodynamic simulations assuming longitudinal boost-invariant.

Cheet sheet for normalization factors in `Trento`,

Collision sytem	Collision energy (GeV)	norm factor	cross section (mb)
Au+Au	200	5.7	42
Pb+Pb	2760	13.9	62
Pb+Pb	5020	18.4	67

Available nucleus type in `Trento`: p, d, Cu, Xe, Au, Pb, U

Finally, the user can specify a random seed for the entire simulation. This is specified inside the block `<Random>` in the xml file. If the `<seed>` parameter is set to 0. Then the random seed will be determined by the system time. If `<seed>` is set to any positive number, the JETSCAPE will perform simulations with the given positive number is the random seed for all its modules. A fix seed simulation will be handy when we study the effect of viscosity during the hydrodynamic evolution.

After each run, please create a result folder with the viscosity information and move the following three files into the result folder,

- `eccentricities_evo_eta_-0.5_0.5.dat`

This file records the evolution of spacial eccentricity of the fireball. Format: `# tau(fm) ecc_n(cos) ecc_n(sin) (n=1-6)`

- `momentum_anisotropy_eta_-0.5_0.5.dat`

This file has the evolution information about the momentum anisotropy, average velocity, and average temperature. Format:

```
# tau(fm) epsilon_p(ideal) epsilon_p(shear) epsilon_p(full) ecc_2 ecc_3 R_Pi gamma T[GeV]
```

- `evolution_for_movie_xyeta_MUSIC.dat`

This file contains the evolution history of fluid cells above $T = 130$ MeV. Every fluid cell includes the following information:

```
itau ix iy ieta volume[fm^4] e[GeV/fm^3] rho_B[1/fm^3] T[GeV] mu_B[GeV] u^x u^y \tau*u^\eta T^{\tau t}[GeV/fm^3] J^\tau[1/fm^3]
```

[Bonus] 5. Produce hadrons from hydrodynamics

In JETSCAPE, a third-party particle sampler iSpectraSampler (iSS) is employed to convert fluid cells to particles. The iSS produces Monte-Carlo particles from the hydrodynamic hyper-surface. The spatial and momentum distributions of particles follow the Cooper-Frye Formula.

The sampled hadrons are output in the `test_out.dat` file. One can run another code script `FinalStateHadrons` to extract the final state hadron list from the `test_out.dat` file. The users need to type in the following command at the build folder,

```
cd ~/JETSCAPE/build
./runJetscape hydro_session/jetscape_user_MUSIC_and_iSS.xml
./FinalStateHadrons test_out.dat hadron_list.dat
```

```

jetscape_user_MUSIC_and_ISS.xml x
1 <?xml version="1.0"?>
2
3 <jetscape>
4   <!-- General settings -->
5   <nEvents> 10 </nEvents>
6   <setReuseHydro> true </setReuseHydro>
7   <nReuseHydro> 10 </nReuseHydro>
8   <JetScapeWriterAscii> on </JetScapeWriterAscii>
9
10  <Random>
11    <seed>0</seed>
12  </Random>
13
14  <!-- Initial State Module -->
15  <IS>
16    <!-- set boost invariant simulation -->
17    <grid_max_z> 0.0 </grid_max_z>
18
19    <!-- Initial condition -->
20    <Trento>
21      <PhysicsInputs projectile='Pb'
22        target='Pb'
23        sqrts='5020'
24        cross-section='6.7'
25        normalization='18.4'>
26      </PhysicsInputs>
27      <CutInputs centrality-low='20'
28        centrality-high='30'>
29      </CutInputs>
30    </Trento>
31  </IS>
32
33  <!-- Preequilibrium Dynamics Module -->
34  <Preequilibrium>
35    <NullPreDynamics> </NullPreDynamics>
36  </Preequilibrium>
37
38  <!-- Hydro Module -->
39  <Hydro>
40    <MUSIC>
41      <name>MUSIC</name>
42      <shear_viscosity_eta_over_s>0.15</shear_viscosity_eta_over_s>
43      <freezeout_temperature>0.150</freezeout_temperature>
44    </MUSIC>
45  </Hydro>
46
47  <!-- Particlization Module -->
48  <SoftParticlization>
49    <ISS> </ISS>
50  </SoftParticlization>
51 </jetscape>
52

```

Oversample 10 particlization events on one hydrodynamic hyper-surface

Add a Cooper-Fry Particlization module

```

jetscape-user@3871e6d29f2c:~/JETSCAPE/build$ ./FinalStateHadrons test_out.dat hadron_list.dat
[Info] 147MB Open Input File = test_out.dat
[Info] 147MB File opened
[Info] 147MB Current Event = 0
Number of hadrons is: 6256
[Info] 150MB Current Event = 1
Number of hadrons is: 6280
[Info] 151MB Current Event = 2
Number of hadrons is: 6611
[Info] 151MB Current Event = 3
Number of hadrons is: 6460
[Info] 151MB Current Event = 4
Number of hadrons is: 6448
[Info] 151MB Current Event = 5
Number of hadrons is: 6548
[Info] 151MB Current Event = 6
Number of hadrons is: 6585
[Info] 151MB Current Event = 7
Number of hadrons is: 6509
[Info] 151MB Current Event = 8
Number of hadrons is: 6417
[Info] 151MB Current Event = 9
Number of hadrons is: 6631
jetscape-user@3871e6d29f2c:~/JETSCAPE/build$

```

With the produced `hadron_list.dat` file, the users can apply their own analysis script to compute particle spectra, mean P_T , and anisotropic flow coefficients v_n . Example analysis codes are `analyze_particle_spectra_and_vn.ipynb` and `analyze_particle_spectra_and_vn.py`.

```

jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ python3 analyze_particle_spectra_and_vn.py
Read in total 10 events.
analyze_particle_spectra_and_vn.py:94: RuntimeWarning: invalid value encountered in true_divide
  v2diff_real_bin_y_ = v2diff_real_bin_y_/pTSpectra_bin_y_
analyze_particle_spectra_and_vn.py:95: RuntimeWarning: invalid value encountered in true_divide
  v2diff_imag_bin_y_ = v2diff_imag_bin_y_/pTSpectra_bin_y_
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$ ls
README.md                               TestRun_avgT_evo.pdf                hydro_evo-ViscosityComp.py
RunCollisionSystemComp_avgv_evo.pdf     TestRun_avgv_evo.pdf                hydro_movie-TempDepVisc.ipynb
RunCollisionSystemComp_ecc2_evo.pdf     TestRun_ecc_evo.pdf                 hydro_movie-TempDepVisc.py
RunCollisionSystemComp_momentumaniso_evo.pdf TestRun_momentumaniso_evo.pdf       hydro_movie-TestRun.ipynb
RunCollisionSystemsComp_avgT_evo.pdf    analyze_particle_spectra_and_vn.ipynb hydro_movie-TestRun.py
RunTempDepVisc_Temperature_Contour_TauX.pdf analyze_particle_spectra_and_vn.py  jetscape_user_AuAu200.xml
RunTempDepVisc_Temperature_Contour_XY.pdf animation_TwithFlow.mp4            jetscape_user_MUSICTestRun.xml
RunTempDepVisc_animation_T.mp4         animation_temperature.mp4          jetscape_user_MUSIC_and_iSS.xml
RunTempDepVisc_animation_TwithFlow.mp4 collect_results.sh                  jetscape_user_PbPb5020.xml
RunViscosityComp_avgT_evo.pdf          figs                                 jetscape_user_TempDepVis.xml
RunViscosityComp_avgv_evo.pdf          hydro_evo-CollisionSystemsComp.ipynb jetscape_user_shear.xml
RunViscosityComp_ecc2_evo.pdf          hydro_evo-CollisionSystemsComp.py  jetscape_user_shear_and_bulk.xml
RunViscosityComp_momentumaniso_evo.pdf hydro_evo-TestRun.ipynb            pTSpectra.pdf
TestRun_Temperature_Contour_TauX.pdf   hydro_evo-TestRun.py                v2_pTdiff.pdf
TestRun_Temperature_Contour_XY.pdf     hydro_evo-ViscosityComp.ipynb
jetscape-user@3871e6d29f2c:~/JETSCAPE/build/hydro_session$

```

HOMEWORK

1. Study the viscous effects on hydrodynamic evolution

Simulate a Pb+Pb collision in 20-30% centrality at 5.02 TeV. In order to compare simulations with different viscosity, we need to fix the random seed. You can choose your favorite number for the seed. (42?)

- Compare the development of flow anisotropy as a function of proper time with two different shear viscosities
- Compare the development of averaged radial with and without the specific bulk viscosity
- Compare the averaged temperature evolution with two choices of the specific bulk viscosity ($\zeta/s(T)$)

2. Produce animation for the temperature and flow velocity profiles in the transverse plane

- Pick your favorite collision system (colliding nuclei, collision energy, and centrality) and generate a hydrodynamic evolution file.

You can try different color maps or even define your own to make the animation vivid.

Please send your best animation to chunshen@wayne.edu. We will select the most impressive ones and post them on the school website.

3. [Bonus] Compute particle spectra and flow anisotropic flow Qn vectors from the event-by-event simulations for one heavy-ion collision system.

To accumulate statistic, you can set `<nEvents>` to 50 and `<nReuseHydro>` to 50 in the xml file to avoid running 50 hydrodynamic simulations. With the generated `test_out.dat` file, apply `FinalStateHadron` and analysis the output to get p_T -spectra for charged hadrons and their flow anisotropy coefficients.