SMASH hands-on session

Dmytro Oliinychenko
JETSCAPE summer school
15-16 July 2020

Goals

- Use SMASH as a hadronic afterburner
- Generate ROOT output of particles and collisions
- By analyzing it learn about chemical and kinetic freeze-out
- (Bonus) Generate SMASH vtk output and look at visualizations of collisions

Start and prerequisites

1. All step by step instructions are on the summer school github

   https://github.com/doliinychenko/SummerSchool2020/tree/master/SMASH_session

2. Add yourself to the table to track progress
3. Are you ready with prerequisites?
   a. Try

      ```
      docker start -ai myJetscape
      ```

   b. Outside of docker environment try

      ```
      root -l
      new TBrowser
      ```

   c. If ROOT is not installed, it's ok, there are alternative instructions for this case

Getting SMASH ready

1. Start the docker environment

   ```
   docker start -ai myJetscape
   ```

2. Compile SMASH

   ```
   cd jetscape-docker/JETSCAPE/external_packages
   ./get_smash.sh
   cd smash/smash_code/build
   make smash -j2
   ```

3. Try if smash runs

   ```
   ./smash --help
   ./smash --version
   ./smash
   ```

Configuring SMASH (1)

Way 1: by config file (by default it's config.yaml)

```
cd JETSCAPE/external_packages/smash/smash_code/build
cp config.yaml JETSCAPE_school.yaml
./smash --inputfile JETSCAPE_school.yaml
```

Way 2: command line options override the config options

```
./smash --inputfile JETSCAPE_school.yaml \
        --config "General: {End_Time: 40.0}"
```

Configuring SMASH (2): looking inside config file

```
Version: 1.8 # minimal SMASH version to use with this config file

Logging:
    default: INFO

General:
    Modus:          Collider
    Time_Step_Mode: Fixed
    Delta_Time:     0.1
    End_Time:       200.0
    Randomseed:     -1
    Nevents:        1

Output:
    Output_Interval: 10.0
    Particles:
        Format:          ["Oscar2013"]

Modi:
    Collider:
        Projectile:
            Particles: {2212: 79, 2112: 118} #Gold197
        Target:
            Particles: {2212: 79, 2112: 118} #Gold197

        E_Kin: 1.23
        Fermi_Motion: "frozen"
```

Looking at the output in ASCII format

By default SMASH output will be in the folders data/0, data/1, etc. Open the latest data/* folder and look at the files there.

```
#!OSCAR2013 particle_lists t x y z mass p0 px py pz pdg ID charge
# Units: fm fm fm fm GeV GeV GeV GeV GeV none none e
# SMASH-1.8
# event 1 out 2115
100 -86.066 20.0817 -43.3986 0.138 1.19694192 -1.05057152 0.26059661 -0.49195004 111 0 0
100 30.3854 -18.1492 85.2682 0.138 0.38667939 0.13206396 -0.05721543 0.33130418 111 1 0
100 78.8696 1.37451 54.5189 0.138 0.60013292 0.47846029 -0.01712481 0.33451161 111 2 0
100 26.5076 54.2063 -78.1123 0.138 0.86598482 0.25917814 0.46049726 -0.67205259 111 3 0
100 -35.7398 4.71614 -90.5283 0.138 0.60341995 -0.21067676 0.00980892 -0.54826155 111 4 0
100 -25.883 -24.7854 -91.94 0.138 1.17842194 -0.29490499 -0.29357803 -1.09383603 111 5 0
100 6.58879 -70.8482 -40.0938 0.138 0.25237496 0.02533322 -0.18771436 -0.09365185 111 6 0
100 56.1153 -51.8371 -39.5174 0.138 0.2641728 0.15208277 -0.12690684 -0.10727886 111 7 0
100 -71.3124 -53.7062 32.7149 0.138 0.89655798 -0.65722592 -0.5181587 0.2904098 111 8 0
100 -13.1515 32.8023 88.8509 0.138 0.51895719 -0.07585658 0.16435041 0.46637677 111 9 0
100 31.687 17.9561 88.6512 0.138 1.60563831 0.64129562 0.33017107 1.42785062 111 10 0
100 4.61924 1.22062 -96.7752 0.138 0.68075515 0.02002094 -0.00538217 -0.66629855 111 11 0
```

Generating output in Root format

config_SMASH_tutorial_collider.yaml

```yaml
Version: 1.8 # minimal SMASH version to use with this config file

Logging:
  default: INFO

General:
  Modus:          Collider
  Time_Step_Mode: Fixed
  Delta_Time:     0.1
  End_Time:       200.0
  Randomseed:     -1
  Nevents:        50

Output:
  Output_Interval: 10.0
  Particles:
      Format:          ["Oscar2013", "Root"]

Modi:
  Collider:
      Projectile:
          Particles: {2212: 79, 2112: 118} #Gold197
      Target:
          Particles: {2212: 79, 2112: 118} #Gold197

      E_Kin: 1.23
      Fermi_Motion: "frozen"
```

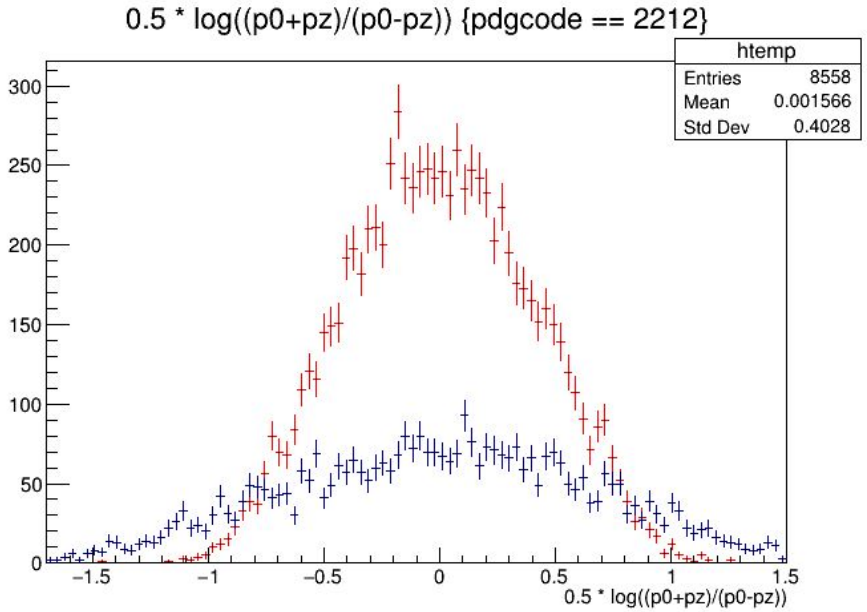
Run SMASH: ./smash --inputfile config_SMASH_tutorial_collider.yaml

Analyzing the ROOT output

root -l

```
TFile *f=new TFile("data/1/Particles.root");
TTree *particles=(TTree*)f->Get("particles");
particles->Scan("*");

particles->Draw("0.5 * log((p0+pz)/(p0-pz))","pdgcode == 2212", "E");
c1->SaveAs("Rapidity_spectrum_protons.png");

particles->Draw("0.5 * log((p0+pz)/(p0-pz))",
                "pdgcode == 211 || pdgcode == 111 || pdgcode == -211", "E");
c1->SaveAs("Rapidity_spectrum_pions.png");

particles->Draw("0.5 * log((p0+pz)/(p0-pz))","pdgcode == 2212", "E");
particles->Draw("0.5 * log((p0+pz)/(p0-pz))",
                "pdgcode == 211 || pdgcode == 111 || pdgcode == -211", "E same");
htemp->SetLineColor(kRed);
c1->SaveAs("Rapidity_spectra_comparison.png");
```

Comparing pion and proton rapidity spectra



Pions, protons (press yes) OR Pions, protons (press no)?

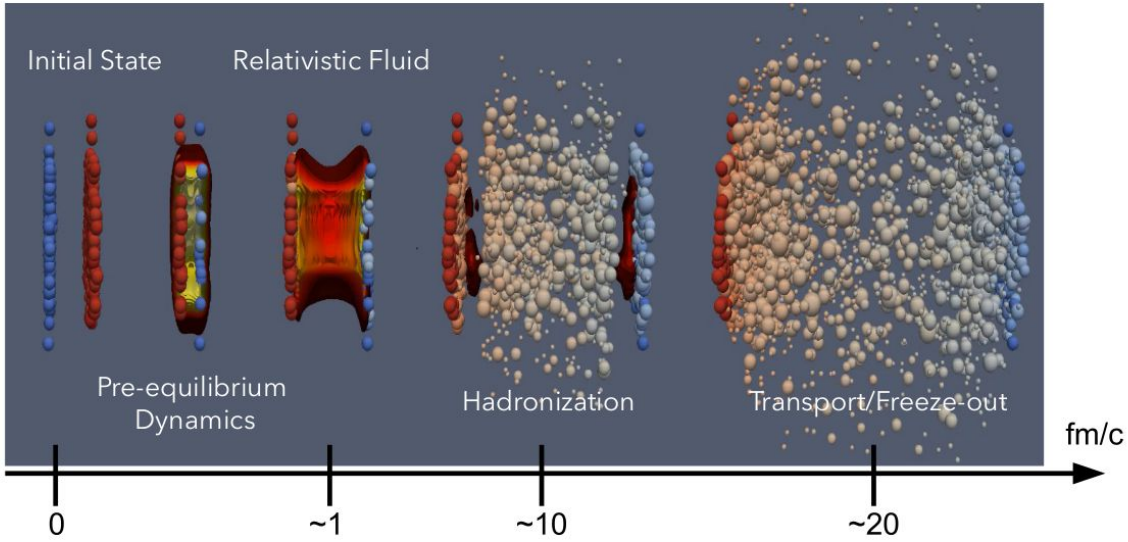
Homework for tomorrow

Install paraview

see instructions at
https://github.com/doliinychenko/SummerSchool2020/tree/master/SMASH_session

Part 2: exploring chemical and kinetic freeze-out

Running SMASH as a hadronic afterburner



Hydrodynamics ➔ Sampler ➔ (I prepared sampled particles from this stage) ➔ Hadronic afterburner

Download sampled particles

Central Au+Au collision at 19.6 GeV, smooth initial condition, MUSIC hydrodynamics, iSS sampler, energy density at particlization 0.26 GeV/fm^3.

1. Download the SMASH_input_particles_from_MUSIC_hydro.tar.gz, see instructions for the link
2. cd JETSCAPE/external_packages/smash/smash_code/build
3. tar -xvf SMASH_input_particles_from_MUSIC_hydro.tar.gz
4. You should get the file sampled_particles0

Prepare config and run SMASH

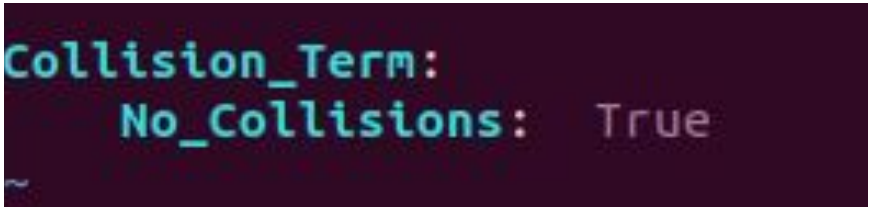
```
Version: 1.8 # minimal SMASH version to use with this config

Logging:
    default: INFO

General:
    Modus:          List
    End_Time:       100.0
    Nevents:        100
    Randomseed:     -1

Output:
    Output_Interval:  100.0
    Particles:
        Format:     ["Root"]
        Extended:   True
        Only_Final: No
    Collisions:
        Format:     ["Root"]
        Extended:   True

Modi:
    List:
        File_Directory: "."
        File_Prefix:    "sampled_particles"
        Shift_Id:       0
```

config_SMASH_tutorial_afterburner.yaml

./smash --inputfile config_SMASH_tutorial_afterburner.yaml

In another tab: run SMASH without collisions

config_SMASH_tutorial_afterburner_no_collisions.yaml



./smash --inputfile config_SMASH_tutorial_afterburner_no_collisions.yaml

Let’s discuss what we are doing

1. Press yes/no if you know what chemical/kinetic freeze-outs are. Think for 2 minutes what are the definitions of chemical and kinetic freeze-outs. In 2 minutes write your definitions in the chat.
2. We ran SMASH as an afterburner with and without collisions. What information can we extract from these runs and how? Suggest your ideas. Think for 2 minutes and then post in chat.

   Bonus: suggest ideas for a paper based on such simulations.

3. What can we potentially say about chemical and kinetic freeze-out based on our simulations? Think for 2 minutes, then post.

Analyze the results

You may use some code and ideas from the step by step instructions:

https://github.com/doliinychenko/SummerSchool2020/tree/master/SMASH_session

1. How much does the hadronic rescattering change the yields and spectra?
2. What can you say by looking at collisions?
3. What can you say about chemical freeze-out?
4. What can you say about kinetic freeze-out?

Bonus: pretty visualization using paraview

Install paraview -- open source visualization application -- on your computer

(NOT in docker environment)

See my step by step instructions, or just go to paraview.org

Bonus: pretty visualization using paraview

Generate VTK output with SMASH:

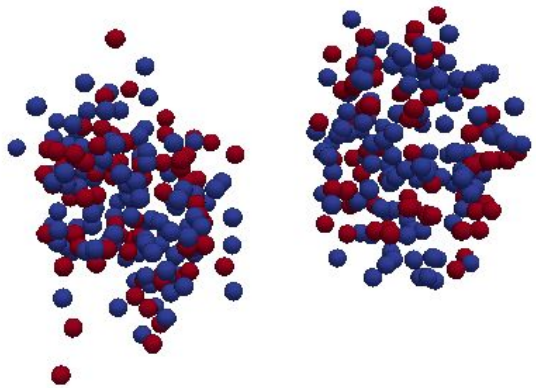
```
General:
    ...
    End_Time:    40.0   # 200.0
    ...
Output:
    Output_Interval: 1.0
    Particles:
        Format:         ["Oscar2013", "VTK"]
        Only_Final:      No
        Extended:        True
```

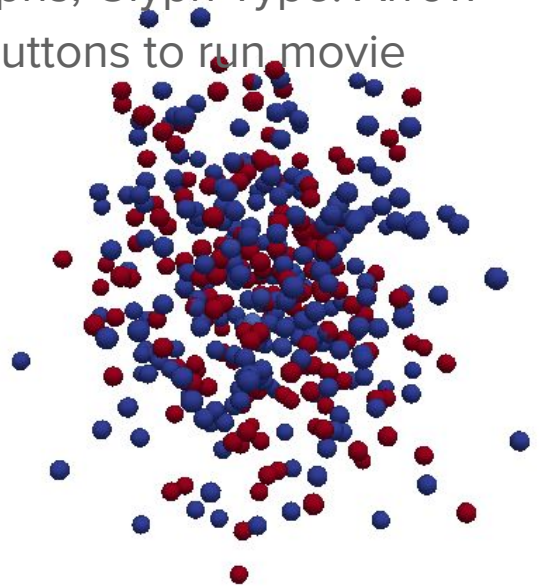
Run SMASH, look at the output folder (latest data/*), you should see many .vtk files

Open them with paraview

Bonus: pretty visualization using paraview

1. Open vtk files with paraview
2. Press large Apply button
3. Change Representation: Surface -> 3D Glyphs, Glyph Type: Arrow -> Sphere
4. Use the Next Frame and Previous Frame buttons to run movie





Feedback

Have you learned anything from the hands on session, press yes/no. No is also ok!

If yes, write 1-2 things in the chat that you actually learned.