CosmoBit

Tutorial

Patrick Stöcker

stoecker@physik.rwth-aachen.de

Institute for Theoretical Particle Physics and Cosmology
RWTH Aachen University

November 04, 2020 – TOOLS 2020

Before you get started

Installing GAMBIT

- For this tutorial you will find the following two papers very helpful
  - The main paper of GAMBIT arXiv:1705.07908
  - The ScannerBit paper arXiv:1705.07959
  - The CosmoBit module paper arXiv:2009.03286
- You need to have version 1.5.1 of GAMBIT. You can either ...
- ... download GAMBIT and build it yourself
  - Go to https://gambit.hepforge.org/downloads/?f=GAMBIT and download the latest version
  - Alternatively, clone from https://github.com/GambitBSM/gambit_1.5
  - Follow the steps in appendix A of the CosmoBit module paper to build gambit and all required backends.
- ... use one of our precompiled docker images (We recommend to do this)
  - Make sure that you have docker installed and that you have the user rights to run it. This may require to add yourself to the `docker` group (If in doubt, `stackexchange` is your friend).
  - Be aware that the image will require quite a lot of disk space (as you are about to get a full installation of GAMBIT and all its default backends).
    Please have about 15-20 GB reserved. The download itself is compressed (~4.5 GB)
  - Get the image by executing
    ```
    docker pull gambitbsm/gambit-pippi:1.5.1-ubuntu-py2
    ```
  - After yopu pulled the image, execute it with
    ```
    docker run -it gambitbsm/gambit-pippi:1.5.1-ubuntu-py2
    ```
- In either case, be aware that it may take some time to build gambit / download the docker image. Please make sure to get everything ready ahead of time.

Using the GAMBIT diagnostics system

- Before we get started let's have a look on the diagnostics system of GAMBIT
- Let's check which backends are currently installed
  - Run **`./gambit backends`**, and verify that AlterBBN, classy, DarkAges, MontePythonLike, MultiModeCode, and plc are installed
- The diagnostics system can also be used to get a short description for a given capability and a list of the functions / modules that provide this capability by running **`./gambit <capability>`**
  - Run **`./gambit H0`**. You should see that there are two possibilities to get H_0. It can either be directly taken from the `LCDM` model or it can be used as a derived quantity of the classy backend
- Similarly, the diagnostics system can also be used to get a short description of a given model and its parameters by running **`./gambit <model>`**
  - Run **`./gambit LCDM`** and **`./gambit LCDM_theta`**. How does this model differ from the definition of ΛCDM that you are accustom to? What is the difference between `LCDM` and `LCDM_theta`?
- For further information on the diagnostics system, please have a look on section 10.4 of the main GAMBIT paper.

Let's use GAMBIT

Choose your own adventure

- To run a scan with gambit the command is:

```
./gambit (-d) (-r) -f <input_file.yaml>
```

- The `-d` flag can be used to initiate a dry run. This can be done to validate whether your input file leads to an unique dependency tree
- When using the `-r` flag, you will restart the scan and delete all previous outputs
- You can choose between three different tutorials (You can also do all of them, if you like)
  - If you just want to get a quick overview of what CosmoBit has to offer and you want to do a few quick runs for a few parameter points, go for Tutorial A
  - If you are keen to modify an existing input file to perform a new scan, then I suggest to give Tutorial B a go
  - If you want to get your hands dirty with adding a new model and implementing the needed functions, Tutorial C is worth a try
- Please be aware that you need to recompile GAMBIT when you choose Tutorial C. This can take a fair amount of time and requires enough memory. If you are also interested in one of the other two options, consider to do them first

Tutorial A

- Let's start with running **`yaml_files/CosmoBit_quickstart.yaml`**. This will run the scan for two random points and will also print out some useful debug information on screen such as the full set of model parameters and a breakdown of the likelihood contributions that were calculated
- If you want to disable these messages, you have to set **`debug: true`** to **`debug: false`** in the **`KeyValues`** section at the bottom of **`yaml_files/CosmoBit_quickstart.yaml`**
- To increase the number of points, you can increase the value of **`point_number: 2`** to something like 10 in the **`Scanner`** section of the YAML file
- In a realistic scan you would not scan randomly through the parameter space; you would rather use a dedicated scanning algorithm. To this end, you need to exchange the `random` scanner by `multinest` by setting **`use_scanner: random`** to **`use_scanner: multinest`**
- The given settings for `multinest` refer to the settings that were chosen to produce the results shown in figure 5 of the module paper. Running these on your local machine will take a lot of time. Please consider to decrease the settings to something like **`tol: 0.1`** and **`nlive: 200`**
- To finish this low resolution scan in a finite amount of time, consider to run it on multiple MPI processes. For 6 MPI processes run **`mpirun -np 6 ./gambit -f <input_file.yaml>`**.
  The scan will then take about 30 mins to finish

Tutorial B

- Let's start with running **`yaml_files/CosmoBit_tutorial.yaml`**
- This YAML file has a lot of useful comments and will show you all possible inputs that are relevant for the capabilities of CosmoBit. Please feel free to play around with the option and try to run the file again to see how this affects the scan.

- When you feel ready, I want like you to create an input file for a modified scenario of **`yaml_files/CosmoBit_quickstart.yaml`**
  - Create a copy of **`yaml_files/CosmoBit_quickstart.yaml`** and call it **`yaml_files/CosmoBit_tutorial_B.yaml`**
  - All files of the scan should be redirected to **`runs/CosmoBit_tutorial_B`**. Can you please change this?
  - So far, there is no $\Delta N_{\rm eff}$ assumed. I would like you to include a model that represents some dark radiation content that is the same at BBN and at recombination. It should be scanned with a flat prior in the range $[0.0, 2.0]$.
  - So far, we assume that the CMB temperature is constant. Can you please change it such that it is scanned over with a Gaussian prior according to the FIRAS measurement $2.72548\,\mathrm{K} \pm 0.00057\,\mathrm{K}$.
  - Now I want you to include the Planck likelihoods. In particular I want you to include the `highl_TTTEEE`, `lowl_TTEE`, and `lensing` likelihood of the 2018 data release. Keep in mind that the Planck likelihoods come with nuisance parameters that need to be scanned with appropriate priors. In the folder **`yaml_files/include/`** you will find helpful files.
  - I would like that the result of MontePython's `sh0es` likelihood is included as an observable but should not be used to drive the scan. Can you help me here?
  - Switch to the MultiNest scanner and reduce the setting to 500 live points and a tolerance of 0.1
  - Now that everything is prepared, disable the debug messages

Tutorial C

- For this tutorial I would simply like you to follow one of the 'how-to' instructions that are provided in the appendix of the module paper. You can choose between ...
  - ... adding a dark energy model as described in the paragraph '**Setting CLASS input for a new model**' of appendix C.2. Please note that this already assumes that the model **DE_model** is defined.
  - ... adding an annihilating dark matter model $\chi\chi \to \pi^0\pi^0$ as outlined in the paragraph '**Implementing a new model of energy injection**' of appendix C.3
- After you have implemented the changes, go to the build folder and run `make gambit`. As pointed out before, this will take some time
- Add the newly-defined models to a YAML file and try to execute it. Keep in mind to choose sensible values for the parameters

If you need help or have some suggestions

- I will be available in zoom room B between 14:30 and 15:30.
- If needed, I will show you the solutions of ‘Tutorial B’. In any case, I will add the solutions to the files on indico shortly before the tutorial.
- If you have questions outside the ‘office hours’, contact me on mattermost or drop a mail (stoecker@physik.rwth-aaachen.de)
- *This is the first time that this CosmoBit tutorial is offered. If you have suggestions on how to improve, please let me know*