Tuesday 29 Jul 2014 Instruction Manual
2014 ISSAC Workshop
E. Grohs & M. Paris

Also available in soft-copy form on `gordon` in:

`/oasis/scratch/egrohs/temp_project/ISSAC_2014/Tue/Tue_Instructions.pdf`

INTRODUCTION

In this workshop we will investigate departures from nuclear statistical equilibrium (NSE) and weak equilibrium in Big Bang Nucleosynthesis (BBN). This instruction manual is to guide you through the process of understanding the relevant physics and computations. Throughout this document, $\hbar = c = k_B = 1$ and the relevant energy unit is MeV. The BBN code is a Fortran90 version of the Wagoner-Kawano code called BURST. Questions are encouraged and should be directed to either Evan Grohs or Mark Paris.

PHYSICS

In BBN, the Boltzmann equation governs the production and destruction of a given nuclear species. If the destruction and production rates of a given nuclear species are rapid compared to any dynamical time scale, then the system will achieve NSE. In the case of BBN, the dynamical time scale is purely gravitational and, as we've learned in the lecture, is set by the Hubble rate, H. If the system achieves NSE, then there is no net production or destruction and the Boltzmann equation vanishes identically. The equilibrium condition in this case is expressed as that of *diffusive equilibrium*, which leads to the relation

$$\mu_A = Z\mu_p + N\mu_n, \tag{1}$$

where μ_i is the chemical potential of species i and the atomic mass number of the nucleus equals the sum of the number of protons and neutrons, i.e. $A = Z + N$. Treating baryons classically as particles that satisfy Maxwell-Boltzmann statistics, we can derive an expression for the chemical potential from the free energy. The expression is:

$$\mu_i = -T \ln \left(\frac{g_i}{n_i} \left(\frac{m_i T}{2\pi} \right)^{3/2} \right) + m_i \tag{2}$$

where

g_i: spin for species i
m_i: mass for species i
n_i: number density of species i
T: plasma temperature.

Substitution of Eqn.(2) for each species into Eqn.(1) yields a relationship between the nuclide number densities, temperature and nuclide masses and spin multiplicities. In an expanding universe, the number density of any species will change due to the time dependence of the metric. This contribution to the time derivative of the number density – the Hubble expansion – can be eliminated by taking the ratio of the number density to any covariantly conserved quantity, such as the baryon number density. The ratio of the given species's number density to the baryon number density, defined as

$$Y_i \equiv \frac{n_i}{n_b}$$

where n_b is the total baryon number density, is the *abundance*. In this way, we can find an expression relating abundances, baryon number density and temperature in NSE.

The equations for weak equilibrium are similar to those of NSE. The relevant reactions are the *Urca (link)* processes:

$$n\nu_e \leftrightarrow pe^-, \qquad ne^+ \leftrightarrow p\bar{\nu}_e, \qquad n \leftrightarrow pe^-\bar{\nu}_e. \tag{3}$$

Akin to the case of NSE and Eqn.(1), weak equilibrium implies the following condition:

$$\mu_n + \mu_{\nu_e} = \mu_p + \mu_{e^-} \tag{4}$$

The electron fraction Y_e is defined as:

$$Y_e = \frac{n_{e^-} - n_{e^+}}{n_b} \tag{5}$$

Using Eqns.(4) and (2) in Eqn.(5), we can find a relationship between Y_e, μ_{ν_e}, μ_{e^-} and temperature in weak equilibrium.

COMPUTATION

1. Log into gordon using ssh:
   `> ssh -XY [user]@gordon.sdsc.edu`
   where `[user]` is your gordon user name. We must use the `-XY` option for this workshop.

2. If you are not using bash, change your shell to bash:
   `> bash`

3. Purge all of the modules you are currently using:
   `> module purge`

4. Go into whichever directory you want to work in, e.g.
   `> cd /oasis/scratch/[user]/temp_project`.

5. Copy the .tgz file for today into your working directory:
   `> cp /oasis/scratch/egrohs/temp_project/ \`
   `ISSAC_2014/Tue/Tue_29Jul2014.tgz ./`

6. Unpack the .tgz file:
   `> tar -xvf Tue_29Jul2014.tgz`

7. There is now a directory named `Tue_29Jul2014` in your working directory. Go into the directory:
   `> cd Tue_29Jul2014`

8. The current working directory contains the following items:

   (a) Fortran90 source files (`*.f90`)

   (b) Input parameter data files (`*.ini`)

   (c) A makefile (`Makefile`)

(d) A bash script to submit jobs to the queue (`torque_submitter`)

(e) A directory with python scripts to make plots (`./plotting/*.py`)

9. The code will run successfully with these `.f90` and `.ini` files. Compile and submit the code to the queue:

   (a) Load gnu and openmpi modules:
   `> module load gnu openmpi_ib intel`

   (b) Compile the code to create the executable `./burst` using make:
   `> make`
   You should see `mpif90` commands output to the terminal. The compilation is successful if the last command links the `.o` files to create the executable `./burst`.

   (c) Optional: You can change the job name which you submit to the queue. Open `torque_submitter` using your preferred editor[1], *e.g.* `vi`:
   `> vi torque_submitter`
   On line five, change the job name from `myburstjob` to whatever name you desire.

   (d) Submit the job to the queue using `torque_submitter`:
   `> qsub torque_submitter`

   (e) `torque_submitter` will request a single core for twenty seconds to run the executable. You can monitor the status of the job using qstat:
   `> qstat -u [user]`

   (f) Once the job completes, there should now be a file named `stdout.out`, a file named `stderr.err`, and four `.dat` files in your working directory. Look at `equil.dat` by using the `cat` command:
   `> cat equil.dat`
   The file shows columns of data. The first column is the plasma temperature (in MeV), and columns two through five show the NSE abundances. The abundances are all zero. You can also use `cat` to look at `ye.dat`, which shows plasma temperature, equilibrium electron fraction, and actual electron fraction. The electron fractions are zero.

10. We now edit the source code so that we print out the NSE abundances.

    (a) Before you edit the source code, derive an expression for the NSE abundance using the equations in the previous section. You should have a term which involves the ratio of the masses. Take $m_p = m_n = m_A/A = m_b$ where m_b is the atomic mass unit. Your result should have a factor involving the difference of masses in an exponent, i.e. $\exp[-(m_A - Zm_p - Nm_n)/T]$. *Do not* take the mass difference to be zero in this term. Use the binding energy number to describe this term.

    (b) Open `nse.f90` using your preferred editor.

    (c) Go to line 72.

    (d) The line currently says `ya(j) = 0._dl`. Delete the zero-number expression and put in the expression for the abundance Y_a deduced from chemical equilibrium. All relevant quantities are saved in the `nse` module, or explicitly input or computed in the `equil` subroutine. The only exception is the number π. That exists in a different module which `nse` uses. The variable is simply called `pi`.

[1] Regarding text editors: we encourage you to learn `vi`/`vim` but `emacs`, `gedit`, etc. are available on `gordon`, too. A quick introduction to `vi`, which should take less than 10 minutes, is available from 'An extremely quick and simple introduction to the vi text editor' (link).

(e) Once you have written the NSE abundance expression, save the changes, recompile using `make`, and resubmit the job to the queue using `torque_submitter`, changing the job name if you desire. When the job runs, it will automatically delete the old copies of `*.dat` in the working directory. If you want to save those copies to compare with the new output, rename the files before submitting the job to the queue.

(f) Check `equil.dat` to ensure the data writes properly.

11. Make a plot of the abundances changing with the plasma temperature, comparing NSE to the actual Boltzmann integration.

    (a) Copy `equil.dat` and `bbn.dat` into `plotting/`:
    ```
    > cp equil.dat plotting/
    > cp bbn.dat plotting/
    ```

    (b) Load the modules to run python:
    ```
    > module purge
    > module load intel gnubase python
    ```

    (c) Go into the plotting directory and execute the python script `abund_v_temp.py`:
    ```
    > cd plotting
    > python abund_v_temp.py
    ```
    This will make a .pdf called `plot_abund_v_temp.pdf`.

    (d) Secure copy the .pdf back to your home machine to view the plot. Bring up a new terminal on your home machine and copy the file from the directory on gordon, e.g.:
    ```
    > scp [user]@gordon.sdsc.edu:/oasis/scratch/[user]/temp_project/ \
    Tue_29Jul2014/plotting/plot_abund_v_temp.pdf ./
    ```

12. Now change the file `ye.f90` to print non-zero electron fractions. Derive an expression for Y_e in weak equilibrium using the equations of the previous section. Ignore the chemical potentials of the leptons, i.e. $\mu_{\nu_e} = \mu_{e^-} = 0$ since we are assuming weak equilibrium. Open `ye.f90`, and change lines 54 and 57. For the actual electron fraction, derive the expression using the total neutron and proton abundances. All quantities are in the `ye` module in `ye.f90`. Save, reload modules, recompile, rename, and resubmit.

13. Once the file `ye.dat` outputs, copy the file into `plotting/`. Reload the modules relevant to python and create the electron fraction plot:
    ```
    > python ye_v_temp.py
    ```
    The output plot is `plot_ye_v_temp.pdf`. Secure copy the `.pdf` back to your home machine to view.

EXTRA CREDIT

There is a discrepancy in the NSE abundance vs. actual abundance of ^{4}He at high temperatures. The discrepancy has nothing to do with the plotting tools. Furthermore, the NSE abundance is correct. The discrepancy is due to the computation. Those who are curious should ask the instructors for further information.