

# AlterBBN tutorial

**Alexandre Arbey**

Online@Lyon

AA, Jérémy Auffinger, Kevin Hickerson & Espen Jenssen  
arXiv:1106.1363 and 1806.11095

<https://alterbbn.hepforge.org/>

**TOOLS 2020**

**Online@Lyon – November 4th, 2020**

## Main purpose

Computing the abundances of the elements predicted by Big-Bang nucleosynthesis (BBN)

## Main features

- public and open-source C program
- (hopefully) easy to modify and user-friendly
- includes different cosmological scenarios
- possible to incorporate simple WIMPs
- parallel computing of uncertainties
- no external library needed
- different solvers for fast-and-dirty to snail-slow-and-very-precise calculations
- automatic uncertainty and correlation calculations
- possibility to extend the nuclear network via automatic interface to REACLIB

## Webpage

Download from

`https://alterbbn.hepforge.org/`

A detailed manual is also available.

## Compilation instructions

- to uncompress: `tar xzvf alterbbn_v2.2.tgz`
- enter directory: `cd alterbbn_v2.2/`
- if you do not want to use gcc, modify compiler in Makefile
- to compile the library: `make`
- to compile the program `xxx.c`: `make xxx`

### Main programs

- `stand_cosmo.c`: standard cosmological model predictions
- `alter_eta.c`: modification of the baryon-to-photon ratio
- `alter_neutrinos.c`: modification of the number of neutrinos
- `alter_neutron.c`: modification of neutron lifetime
- `alter_etannutau.c`: modification of the above parameters
- `alter_standmod.c`: cosmological scenario with dark energy and/or dark entropy
- `alter_reheating.c`: cosmological scenario with radiation entropy injection
- `alter_phi.c`: cosmological scenario with scalar field decaying to radiation
- `alter_wimps.c`: cosmological scenario with light WIMPs

### With which arguments?

Running `./program.x` without argument will show you the possible arguments

## A first example

```
./stand_cosmo
```

```
failsafe 0 = fast  
1..3 = more precise, stiff method  
5..7 = stiff method with precision tests (5=tolerance of 5%, 6=tolerance of 1%, 7=tolerance of 0.1%)  
10..12 = RK4 method with adaptative stepsize (10=5%, 11=1%, 12=0.1%)  
20..22 = Fehlberg RK4-5 method (20=5%, 21=1%, 22=0.1%)  
30..32 = Cash-Karp RK4-5 method (30=1%, 31= $10^{-4}$ , 32= $10^{-5}$ )
```

## A first example

```
./stand_cosmo 1
```

	Yp	H2/H	He3/H	Li7/H	Li6/H	Be7/H
low:	2.470e-01	2.508e-05	1.019e-05	5.076e-10	1.673e-15	4.794e-10
cent:	2.469e-01	2.427e-05	1.030e-05	5.478e-10	1.067e-14	5.193e-10
high:	2.469e-01	2.387e-05	1.039e-05	5.796e-10	3.490e-14	5.506e-10

-----  
With uncertainties:

	Yp	H2/H	He3/H	Li7/H	Li6/H	Be7/H
value:	2.469e-01	2.427e-05	1.030e-05	5.478e-10	1.067e-14	5.193e-10
+/- :	3.209e-04	9.058e-07	1.814e-07	4.335e-11	1.065e-14	4.273e-11

Correlation matrix:

	Yp	H2/H	He3/H	Li7/H	Li6/H	Be7/H
Yp	1.000000	0.038117	0.007206	-0.001576	0.003166	-0.003881
H2/H	0.038117	1.000000	-0.622503	-0.599145	0.028641	-0.616747
He3/H	0.007206	-0.622503	1.000000	0.403211	-0.014964	0.416617
Li7/H	-0.001576	-0.599145	0.403211	1.000000	-0.018426	0.997743
Li6/H	0.003166	0.028641	-0.014964	-0.018426	1.000000	-0.019078
Be7/H	-0.003881	-0.616747	0.416617	0.997743	-0.019078	1.000000

chi2=2.289

Compatible with BBN constraints (chi2 without correlations)

chi2=2.677

Compatible with BBN constraints (chi2 including correlations)

low/high = values obtained by taking extremal values of nuclear reaction cross sections

### Observational constraints (from PDG)

$$Y_p = 0.2450 \pm 0.003$$

$$[{}^2\text{H}]/[\text{H}] = (2.569 \pm 0.027) \times 10^{-5}$$

$$[{}^3\text{He}]/[{}^2\text{H}] = (1.1 \pm 0.2) \times 10^{-5}$$

$$[{}^7\text{Li}]/[\text{H}] = (1.6 \pm 0.3) \times 10^{-10}$$

Only the helium mass fraction  $Y_p$  and the deuterium abundance  $[{}^2\text{H}]/[\text{H}]$  are used for the  $\chi^2$ , this can be changed in `src/bbn_constraints.c`.

## Which integration solver to choose?

The most precise method is number 12 (Runge-Kutta of order 4 with adaptive stepsize)

failsafe	$Y_p(\times 10^{-1})$	deviation	${}^2\text{H}/\text{H}(\times 10^{-5})$	deviation	${}^7\text{Li}/\text{H}(\times 10^{-10})$	deviation
0	2.462	-0.010	2.343	-0.120	5.680	+0.310
1	2.472	0.000	2.431	-0.032	5.473	+0.103
2	2.472	0.000	2.454	-0.009	5.402	+0.032
3	2.472	0.000	2.459	-0.004	5.382	+0.012
5	2.524	+0.048	2.672	+0.209	4.974	-0.396
6	2.475	+0.003	2.483	+0.020	5.316	-0.054
7	2.472	0.000	2.462	-0.001	5.372	+0.002
10	2.510	+0.038	2.787	+0.314	4.654	-0.716
11	2.476	+0.004	2.488	+0.025	5.304	-0.066
12	2.472	—	2.463	—	5.370	—
20	2.488	+0.016	2.606	+0.143	4.988	-0.382
21	2.479	+0.007	2.521	+0.058	5.212	-0.158
22	2.473	+0.001	2.467	+0.004	5.361	-0.009
30	2.489	+0.017	2.619	+0.146	4.969	-0.401
31	2.477	+0.005	2.488	+0.025	5.311	-0.059
32	2.473	+0.001	2.470	+0.007	5.351	-0.019



## Which integration solver to choose?

Computing time for the central values of the abundances of the elements on 4 cores

failsafe	gcc 8.3	clang 7.0	icc 19.0	gcc w/o OpenMP
0	0.0292439 s	0.0371051 s	0.035774 s	0.030885 s
1	0.0598671 s	0.0702951 s	0.04901 s	0.0663319 s
2	0.12691 s	0.14824 s	0.105646 s	0.139108 s
3	0.513578 s	0.595697 s	0.426564 s	0.561526 s
5	0.241128 s	0.279304 s	0.201433 s	0.260442 s
6	1.83213 s	2.23646 s	1.5879 s	1.97117 s
7	16.9573 s	20.3785 s	14.0737 s	17.8445 s
10	0.843263 s	0.992097 s	0.688307 s	0.89699 s
11	7.81774 s	9.06975 s	6.40297 s	8.35519 s
12	72.1262 s	84.8538 s	59.8514 s	77.7767 s
20	0.157761 s	0.176582 s	0.133258 s	0.169685 s
21	0.54873 s	0.625423 s	0.449118 s	0.597708 s
22	4.84355 s	5.62732 s	4.11268 s	5.26304 s
30	0.200886 s	0.231516 s	0.170542 s	0.215176 s
31	1.53365 s	1.79631 s	1.29438 s	1.68841 s
32	5.85279 s	6.8085 s	4.89614 s	6.41176 s

## Which integration solver to choose?

Computing time for the correlation matrix of the abundances of the elements on 4 cores

failsafe	gcc 8.3	clang 7.0	icc 19.0	gcc w/o OpenMP
0	0.710522 s	0.945136 s	0.618748 s	2.5015 s
1	1.98385 s	2.63015 s	1.73446 s	7.17126 s
2	4.37474 s	6.00581 s	3.70563 s	14.8925 s
3	17.5996 s	22.5279 s	14.915 s	59.1254 s
5	8.0979 s	9.99622 s	6.89687 s	27.2925 s
6	64.6826 s	77.3424 s	51.4998 s	207.582 s
7	549.041 s	721.656 s	467.687 s	1886.54 s
10	27.4793 s	34.2517 s	24.0487 s	94.5529 s
11	256.467 s	321.405 s	218.789 s	882.831 s
12	2344.66 s	3094.74 s	2062.58 s	8312.86 s
20	4.97176 s	7.24952 s	4.54886 s	17.6842 s
21	17.3305 s	28.8574 s	15.2907 s	62.2697 s
22	151.593 s	224.326 s	135.464 s	550.91 s
30	6.28095 s	8.24639 s	5.8179 s	22.2994 s
31	48.7308 s	67.6528 s	44.2791 s	176.329 s
32	188.842 s	258.642 s	175.523 s	671.334 s

Methods 1-3 are good compromises between speed and precision, and 6 for more precision

By default, a nuclear network with 26 elements and 100 nuclear reactions is used, as defined in `src/bbnrate.c`.

index	nuclei	atomic number	charge number	mass excess
1	n	1	0	8.071388
2	p	1	1	7.289028
3	<sup>2</sup> H	2	1	13.135825
4	<sup>3</sup> H	3	1	14.949915
5	<sup>3</sup> He	3	2	14.931325
6	<sup>4</sup> He	4	2	2.424931
7	<sup>6</sup> Li	6	3	14.0864
8	<sup>7</sup> Li	7	3	14.9078
9	<sup>7</sup> Be	7	4	15.7696
10	<sup>8</sup> Li	8	3	20.9464
11	<sup>8</sup> B	8	5	22.9212
12	<sup>9</sup> Be	9	4	11.34758
13	<sup>10</sup> B	10	5	12.05086
14	<sup>11</sup> B	11	5	8.6680
15	<sup>11</sup> C	11	6	10.6506
16	<sup>12</sup> B	12	5	13.3690
17	<sup>12</sup> C	12	6	0
18	<sup>12</sup> N	12	7	17.3382
19	<sup>13</sup> C	13	6	3.125036
20	<sup>13</sup> N	13	7	5.3455
21	<sup>14</sup> C	14	6	3.019916
22	<sup>14</sup> N	14	7	2.863440
23	<sup>14</sup> O	14	8	8.006521
24	<sup>15</sup> N	15	7	0.101439
25	<sup>15</sup> O	15	8	2.8554
26	<sup>16</sup> O	16	8	-4.737036

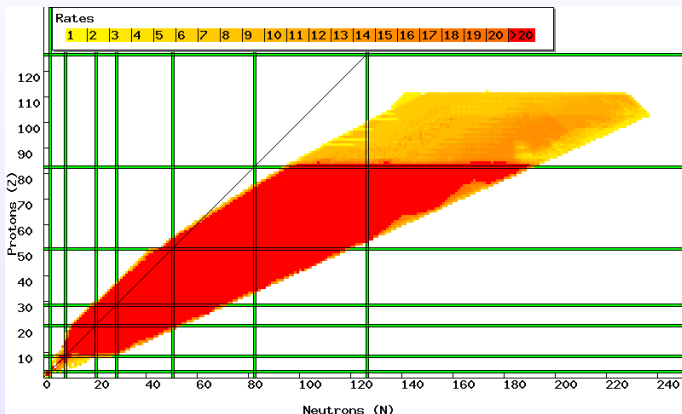
### Have a try with the other main programs:

- `alter_eta.c`: modification of the baryon-to-photon ratio
- `alter_neutrinos.c`: modification of the number of neutrinos
- `alter_neutron.c`: modification of neutron lifetime
- `alter_etannutau.c`: modification of the above parameters
- `alter_standmod.c`: cosmological scenario with dark energy and/or dark entropy
- `alter_reheating.c`: cosmological scenario with radiation entropy injection
- `alter_phi.c`: cosmological scenario with scalar field decaying to radiation
- `alter_wimps.c`: cosmological scenario with light WIMPs

AlterBBN is interfaced with the REACLIB database:

<https://reaclib.jinaweb.org/>

It is huge database of nuclear reactions, which can be used for BBN, supernovae...



### Instructions

- go to `src/contrib/newreac`
- to compile the code: `make`
- `create_network.x` needs 3 parameters:
  - name of the WINVN file
  - name of the REACLIB database file
  - maximal atomic mass (A)
- example: `./create_network.x winvn_v2.0.dat results02200820.dat 30`  
generates a nuclear network with 181 nuclides (up to  $^{30}\text{Ca}$ ) and 909 nuclear reactions  
in practice, `bbn.h`, `bbnrate.h` and `numbers.h` are generated
- to activate the new network in AlterBBN:
  - go back to `/src`
  - open `include.h` and uncomment `//#define REACLIB`
  - go back to the main directory and recompile with `make` and `make stand_cosmo`
  - run `./stand_cosmo.x 1`

Since the nuclear network has been expanded, the computing time also increases.

However, the errors on cross sections not being included in REACLIB, I would advise to disregard the errors for the heavier elements, which are very approximate.

All the main programs are provided as examples, and they can easily be modified to scan over the parameters.

An important line for the output is:

```
H2_H=ratioH[3];Yp=ratioH[6];Li7_H=ratioH[8];Be7_H=ratioH[9];  
He3_H=ratioH[5];Li6_H=ratioH[7];
```

The abundances of the different nuclides are given directly in `ratioH`, which is the ratio of the nuclide density over the hydrogen density

The correspondence between the number and the name of a given nuclide can be found in `src/bbn.h`.

In addition, `ratioH[0]` contains the present value of  $\eta$ , and `ratioH[2]` the H density.

Finally, it is possible to change the nuclear network by hand, in particular in:

- `src/bbnrate.h`: central piece of the nuclear network
- `src/bbn.h`: information about the nuclides

AlterBBN is the result of a collaborative work and is open-source...

So, if you want your modifications to be included in a future version of AlterBBN, do not hesitate to contact us, we are very interested into developing further  
AlterBBN!