Abacus allows the user to choose when and how much data to output. There are five user-defined parameters that control outputs, listed here and described below:

1. **`L1OutputRedshifts`** `= [an array of redshifts (floats)];`
2. **`TimeSliceRedshifts_Subsample`** `= [an array of redshifts (floats)];`
3. **`TimeSliceRedshifts_Full`** `= [an array of redshifts (floats)];`
4. **`ParticleSubsampleBig`** `= a float between 0 and 1;`
5. **`ParticleSubsampleSml`** `= a float between 0 and 1;`

The first three parameters control when and what kind of data to output; the last two control how much data to store. For a given redshift, there are three levels of data volume that can be output. An example is provided on the next page.

1. If the current redshift appears in `L1OutputRedshifts`, but nowhere else, Abacus will output only the smallest dataset (hereafter referred to as Dataset 1). Dataset 1 consists solely of slab-based halo catalogs with statistics for all halos above 50 particles, as well as the particle IDs of a consistent subsample of halo particles. A fixed fraction of the simulation particles are consistently sampled at each epoch and divided into two disjoint and complementary sets; the size of the sets is determined by the parameters `ParticleSubsampleBig` and `ParticleSubsampleSml`. Within these, Abacus identifies subsampled particles belonging to L1 halos and outputs their particle IDs. The particle IDs can be used to associate halos to form merger trees. Note: for redshifts appearing *only* in `L1OutputRedshifts`, Abacus will output Dataset 1 at the first redshift smaller than the next requested `L1OutputRedshift`; Abacus will not shorten its timestep to land precisely on the requested output redshift.

2. If the current redshift appears in `TimeSliceRedshifts_Subsample`, but not in `TimeSliceRedshifts_Full`, Abacus will store all the data for Dataset 1 (this time shortening its timestep to land precisely on the requested output redshift, and including L0 halo particles in the halo subsample particle IDs), as well as additional data for a fraction of the simulation particles, which are chosen consistently at all epochs. The total fraction of the particles for which additional data will be output is given by `ParticleSubsampleBig + ParticleSubsampleSml`. The extra data will include:
   A. the particle IDs of the subsampled field particles, divided into two sets corresponding to the two subsample fractions. This set of files complements the halo particle IDs in Dataset 1. It can be used to relate particle positions to their initial Lagrangian position (as coded in the IDs), which can be used in the derivation of transfer functions.
   B. the positions and velocities of the subsampled particles, separated into two sets corresponding to the two subsample fractions. Each of these sets is then further divided into particles in L0 halos and in the field (particles not belonging to any L0 groups). The halo particles are in the catalog order, and the catalog contains the indexing of the particle files. The combination of field and halo particles gives an unbiased sample of the density field, with enough sampling for most clustering and lensing analyses. Users can opt to access `ParticleSubsampleBig`, `ParticleSubsampleSml`, or their joint set if needed.

3. If the current redshift appears in `TimeSliceRedshifts_Full`, Abacus outputs a set of files containing the positions, velocities, and particle IDs of all of the particles in the simulation, in addition to Dataset 1. For these redshifts, Abacus will shorten its timestep as needed to land precisely on the requested output redshift.

If a redshift appears in more than one of the above redshift array parameters, Abacus will output the highest volume of data requested, avoiding redundancy when necessary. Note: If group finding is turned off (`AllowGroupFinding = 0`), all particles are labeled field particles, whether or not they belong to physical halos.

Example:

```
L1OutputRedshifts             = [6.0, 3.5, 0.5]
TimeSliceRedshifts_Subsample = [4.0, 0.1]
TimeSliceRedshifts_Full      = [3.0]
ParticleSubsampleBig         = 0.03
ParticleSubsampleSml         = 0.07
```

With these settings, Abacus will output the following types of data at the given redshifts:

Output	Path z =	6.0	4.0	3.5	3.0	0.5	0.1
Halo catalogue and statistics	`$GroupDirectory/StepXXXX_zXX.XXX/halo_info_*`	✓	✓	✓	✓	✓	✓
Particle IDs of the 3% particle subsample, halo particles only	`$GroupDirectory/StepXXXX_zXX.XXX/halo_pids_subSml_*`	✓, L1 only	✓, L0 + L1	✓, L1 only	✓, L1 only	✓, L1 only	✓ L0 + L1
Particle IDs of the 7% particle subsample, halo particles only	`$GroupDirectory/StepXXXX_zXX.XXX/halo_pids_subBig_*`	✓, L1 only	✓, L0 + L1	✓, L1 only	✓, L1 only	✓, L1 only	✓ L0 + L1
Particle IDs of the 3% particle subsample, field particles only	`$GroupDirectory/StepXXXX_zXX.XXX/field_pids_subSml_*`		✓				✓
Particle IDs of the 7% particle subsample, field particles only	`$GroupDirectory/StepXXXX_zXX.XXX/field_pids_subBig_*`		✓				✓
Particle IDs of 100% of the particles	`$OutputDirectory/sliceXX.XXX/$SimName.zXX.XXX.slabXXXX_pids.dat`				✓		
Positions and velocities of the 3% particle subsample (separated into halo and field particle files)	`$GroupDirectory/StepXXXX_zXX.XXX/field_rv_subSml_*` and `halo_rv_subSml_*`		✓				✓
Positions and velocities of the 7% particle subsample (separated into halo and field particle files)	`$GroupDirectory/StepXXXX_zXX.XXX/field_rv_subBig_*` and `halo_rv_subBig_*`		✓				✓
Positions and velocities, of 100% of the particles in the box.	`$OutputDirectory/sliceXX.XXX/$SimName.zXX.XXX.slabXXXX.field.dat` and `$SimName.zXX.XXX.slabXXXX.L0.dat`				✓		