Peter Thomas' changes to L-Galaxies

Rob Yates

July 22, 2011

1 New Stuff

- `metals.c`
  - `metals_add`: Transfers metals from one component to another. Either from $M_{\rm cold}$ to M_* in `update_from_star_formation`, or from $M_{\rm cold}$ to $M_{\rm hot}$ or $M_{\rm ejec}$ in `update_from_feedback`.
  - `metals_init`: Initialises the `metals.type1a`, `metals.type2` and `metals.agb` attributes to 0.0 in a galaxy component's `metals` sub-struct.
  - `metals_print`: Prints metals produced by SNe-Ia, SNe-II and AGB stars found in a given galaxy component.
  - `metals_total`: Returns total amount of metals in a given galaxy component (i.e: sum of `type1a`, `type2` and `agb` metals).
- `star_formation_history.c`
  - `sfh_initialise`: Initialises the attributes in the `Gal` struct related to SFH (see §3). `sfh_dt[0]` and `sfh_t[0]` set to 1. Called from `recipe_misc.c`.
  - `sfh_merge`: Checks if two merging galaxies have the same number of bins (i.e: if `Gal[p1].sfh_ibin` = `Gal[p2].sfh_ibin`). If they do, then adds the values for each attribute related to SFH into the main progenitor's `Gal` struct. Also nullify the `sfh_ibin` and `sfh_age` values of the minor progenitor.
  - `sfh_print`: Prints values of attributes in the `Gal` struct related to SFH (see §3).
  - `sfh_update_bins`: Used to update the bin structure in isolated galaxies at each time-step, via a call from `star_formation_and_feedback` (see Fig.1), and to synchronise the bin structures of two merging galaxies (if necessary), via a call from `add_galaxies_together` (see Fig.2).

2 Updated Stuff

- `allvars.h`
  - When STAR_FORMATION_HISTORY is defined, the following constants are added:
    - `SFH_TIME_INTERVAL` = 1.e+7
    - `SFH_NBIN` = 20
  - When METALS is defined, the following struct is defined:

```
struct metals{
    float type1a;
    float type2;
    float agb; }
```

  - When STAR_FORMATION_HISTORY is defined, the following attributes are added to the `Gal` struct:
    - `sfh_ibin`
    - `sfh_age`
    - `sfh_dt[SFH_NBIN]`
    - `sfh_t[SFH_NBIN]`
    - `sfh_StellarMass[SFH_NBIN]`

    Some of these are arrays of size 20.
  - When METALS is defined, `metals`-type sub-structs for the following components are added to the `Gal` struct:
    - `MetalsColdGas`
    - `MetalsStellarMass`
    - `MetalsBulgeMass`
    - `MetalsHotGas`
    - `MetalsEjectedMass`
  - When STAR_FORMATION_HISTORY and METALS are defined, the following `metals`-type sub-struct is added to the `Gal` struct:

    * sfh_MetalsStellarMass[SFH_NBIN]

    This is an array of 20 metals-type sub-structs for the stellar mass component, used for SFH purposes.

- recipe_star_formation_and_feedback.c
  - star_formation_and_feedback: Increases metal content of the $M_{\rm cold}$ component due to 'instantaneously' recycled metals from SNe-II. (Metals moved from cold gas to long-lived stars are calculated in update_from_star_formation).
  - update_from_star_formation: Uses metals_add to update the metals sub-struct for the $M_{\rm cold}$ and M_* components (due to star formation), as well as the attributes in the Gal struct related to SFH (see above).
  - update_from_feedback: Uses metals_add to update the metals sub-struct for the $M_{\rm cold}$ and $M_{\rm hot}$ components (reheating), and then the metals struct for the $M_{\rm hot}$ and $M_{\rm ejec}$ components (ejection).
  - check_disk_instability: Updates the metals transferred to the bulge after a disc instability.
- recipe_mergers.c
  - add_galaxies_together: Firstly, synchronises the bins of the two merging galaxies via sfh_update_bins (could be unsynchronised). Then, merges contents of the bins from each galaxy via sfh_merge.
  - make_bulge_from_burst: Transfers metals from the stellar disc to the bulge via metals_add.
  - collisional_starburst_recipe: Updates the MetalsColdGas.type2 and MetalsHotGas.type2 values due to the 'instantaneous recycling' of SNe-II products into the ISM and hot gas via supernova feedback after a merger-induced starburst.

3 Descriptions

- SFH_TIME_INTERVAL = 1.e+7 = Default 'width' of the SFH bins. Each new sfh_dt element is set to this.
- SFH_NBIN = 20 = Number of bins (i.e: number of elements in the arrays in metals-type structs).
- sfh_ibin = The index of the newest active bin. This would reach 20 just after the present day.
- sfh_age = Age of the universe at last call to sfh_update_bins. Can be that a galaxy merges before forming stars in a given time-step. Then, sfh_age would not be automatically synchronised with the 'normal' galaxy in the merger. add_galaxies_together corrects for this via a special call to sfh_update_bins.
- sfh_dt[SFH_NBIN] = The 'width' of SFH bins in time (in units of SFH_TIME_INTERVAL).
- sfh_t[SFH_NBIN] = Age of the universe at the end of each active bin.
- sfh_StellarMass[SFH_NBIN] = A metals-type struct to contain stellar mass history of a galaxy.

4 To do

- Implement meals produced from star formation in terms of their ejection route (SNe-Ia, SNe-II or AGB stellar winds). Bear in mind that metals need to be dispersed into the gas components *over time* for SNe-Ia and AGB stars. Stellar lifetimes and yield tables needed for this. Look at Arrigoni et al. (2010a), Pipino & Matteucci (2004) and Rob Wiersma's work for how this has been done before.

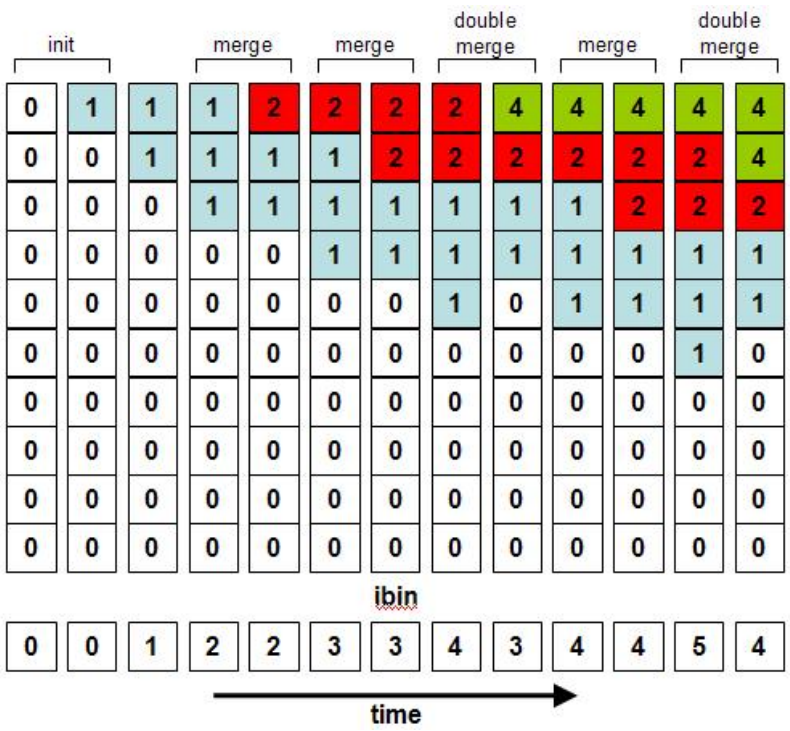

Figure 1: Evolution of the `sfh_dt` array for a non-merging galaxy. The 'width' of each SFH bin (in units of `SFH_TIME_INTERVAL`) is given in each element. Coloured elements are 'active'. When three or more active elements have the same width, two of them are merged. With twenty elements, this process will fill the array just after the present day. The value entered into `sfh_ibin` (the index of the last active bin) is also shown.

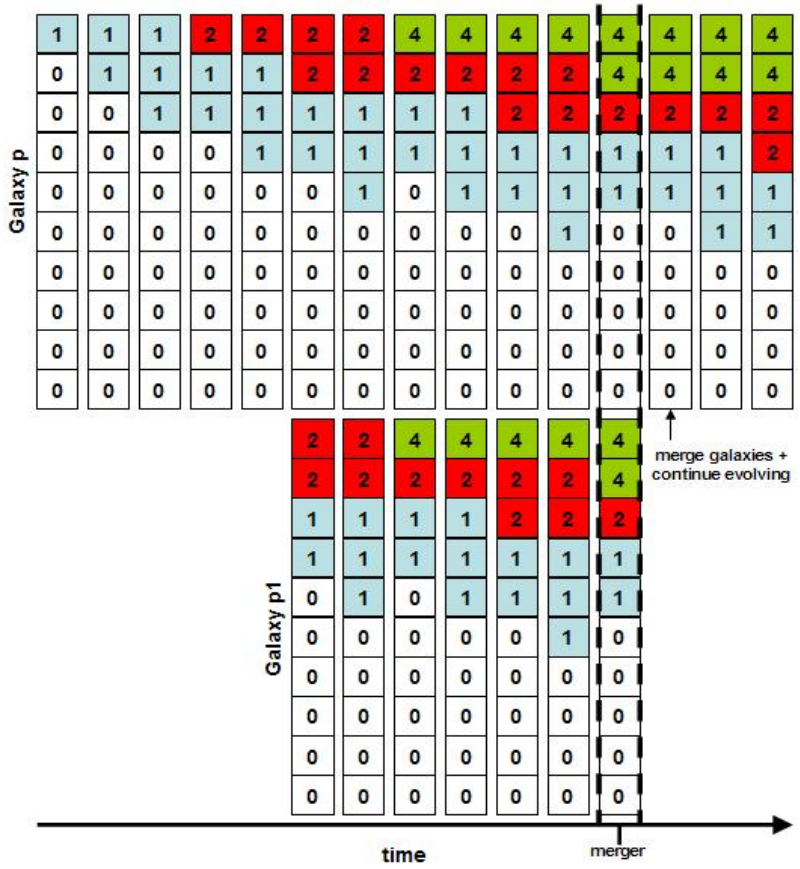

Figure 2: Evolution of the `sfh_dt[NBIN]` array for two merging galaxies. The 'width' of each SFH bin (in units of `SFH_TIME_INTERVAL`) is given in each element. Coloured elements are 'active'. At the time of merger, the bin structures of the two galaxies are compared. If they're the same (should be, but depends on `sfh_age`, see §3), the SFH data can be merged together and the 'descendent' further evolved.