VICE

Release 1.1.0

James W. Johnson

Mar 26, 2020

CONTENTS

1 Science Documentation 3

1.1 Background . . . 3

1.1.1 Galactic Chemical Evolution . . . 3

1.1.2 The Singlezone Approximation . . . 4

1.1.3 The Multizone Approximation . . . 4

1.2 Implemementation . . . 4

1.2.1 Motivation . . . 4

1.2.2 Numerical Approach . . . 4

1.2.3 Minimization of Dependencies . . . 4

1.3 Single Stellar Populations . . . 5

1.3.1 Stellar Lifetimes . . . 5

1.3.2 The Cumulative Return Fraction . . . 6

1.3.3 The Main Sequence Mass Fraction . . . 8

1.4 Enrichment . . . 10

1.4.1 The Enrichment Equation . . . 10

1.4.2 Core Collapse Supernovae . . . 10

1.4.3 Type Ia Supernovae . . . 11

1.4.4 Asymptotic Giant Branch Stars . . . 12

1.4.5 Subsequent Terms . . . 14

1.4.6 Sanity Checks . . . 14

1.5 Nucleosynthetic Yields . . . 15

1.5.1 Core Collapse Supernovae . . . 15

2 User's Guide 17

2.1 From the Command Line . . . 17

2.2 Package Contents . . . 17

2.2.1 vice.cumulative_return_fraction . . . 17

2.2.1.1 Parameters . . . 17

2.2.1.2 Returns . . . 18

2.2.1.3 Notes . . . 18

2.2.1.4 Raises . . . 18

2.2.1.5 Example Code . . . 19

2.2.2 vice.main_sequence_mass_fraction . . . 19

2.2.2.1 Parameters . . . 19

2.2.2.2 Returns . . . 19

2.2.2.3 Notes . . . 19

2.2.2.4 Raises . . . 20

2.2.2.5 Example Code . . . 20

2.2.3 vice.single_stellar_population . . . 20

2.2.3.1 Parameters . . . 20

2.2.3.2 Returns . . . 21
2.2.3.3 Raises . . . 22
2.2.3.4 Example Code . . . 22
2.2.4 vice.imf.kroupa . . . 22
2.2.4.1 Parameters . . . 23
2.2.4.2 Returns . . . 23
2.2.4.3 Raises . . . 23
2.2.4.4 Example Code . . . 23
2.2.5 vice.imf.salpeter . . . 23
2.2.5.1 Parameters . . . 23
2.2.5.2 Returns . . . 23
2.2.5.3 Raises . . . 24
2.2.5.4 Example Code . . . 24
2.2.6 vice.singlezone . . . 24
2.2.6.1 Parameters . . . 24
2.2.6.2 Attributes . . . 24
2.2.6.3 Functions . . . 26
2.2.6.4 Example Code . . . 26
2.2.7 vice.singlezone.from_output . . . 26
2.2.7.1 Parameters . . . 26
2.2.7.2 Returns . . . 27
2.2.7.3 Raises . . . 27
2.2.7.4 Notes . . . 27
2.2.7.5 Example Code . . . 27
2.2.8 vice.singlezone.run . . . 28
2.2.8.1 Parameters . . . 28
2.2.8.2 Returns . . . 28
2.2.8.3 Raises . . . 28
2.2.8.4 Notes . . . 29
2.2.8.5 Example Code . . . 29
2.2.9 vice.singlezone.name . . . 29
2.2.9.1 Example Code . . . 30
2.2.10 vice.singlezone.func . . . 30
2.2.10.1 Example Code . . . 30
2.2.11 vice.singlezone.mode . . . 30
2.2.11.1 Example Code . . . 31
2.2.12 vice.singlezone.verbose . . . 31
2.2.12.1 Example Code . . . 31
2.2.13 vice.singlezone.elements . . . 31
2.2.13.1 Example Code . . . 32
2.2.14 vice.singlezone.IMF . . . 32
2.2.14.1 Example Code . . . 32
2.2.15 vice.singlezone.eta . . . 33
2.2.15.1 Example Code . . . 33
2.2.16 vice.singlezone.enhancement . . . 33
2.2.16.1 Example Code . . . 34
2.2.17 vice.singlezone.entrainment . . . 34
2.2.17.1 Attributes . . . 34
2.2.17.2 Example Code . . . 34
2.2.18 vice.singlezone.Zin . . . 35
2.2.18.1 Example Code . . . 35
2.2.19 vice.singlezone.recycling . . . 35
2.2.19.1 Example Code . . . 36
2.2.20 vice.singlezone.bins . . . 36

2.2.20.1 Example Code 36
2.2.21 vice.singlezone.delay 36
2.2.22 vice.singlezone.RIa 37
2.2.22.1 Example Code 37
2.2.23 vice.singlezone.Mg0 37
2.2.23.1 Example Code 38
2.2.24 vice.singlezone.smoothing 38
2.2.24.1 Example Code 38
2.2.25 vice.singlezone.tau_ia 38
2.2.25.1 Example Code 39
2.2.26 vice.singlezone.tau_star 39
2.2.26.1 Example Code 39
2.2.27 vice.singlezone.dt 40
2.2.27.1 Example Code 40
2.2.28 vice.singlezone.schmidt 40
2.2.28.1 Example Code 41
2.2.29 vice.singlezone.MgSchmidt 41
2.2.29.1 Example Code 41
2.2.30 vice.singlezone.schmidt_index 41
2.2.30.1 Example Code 42
2.2.31 vice.singlezone.m_upper 42
2.2.31.1 Example Code 42
2.2.32 vice.singlezone.m_lower 42
2.2.32.1 Example Code 42
2.2.33 vice.singlezone.postMS 43
2.2.33.1 Example Code 43
2.2.34 vice.singlezone.Z_solar 43
2.2.34.1 Example Code 43
2.2.35 vice.singlezone.agb_model 43
2.2.35.1 Example Code 44

Version 1.1.0

James W. Johnson

Primary Author

Email: giganano9@gmail.com

The Ohio State University Department of Astronomy

140 W. 18th Ave., Columbus, OH, 43204

Welcome to VICE's documentation! Source code and more resources can found in the git repository. To cite this version of VICE, please reference Johnson & Weinberg (2020). Any modifications to the source code will require rebuilding VICE from source for the change to take effect.

CHAPTER

ONE

SCIENCE DOCUMENTATION

In this documentation we adopt the notation where a lower-case m implicitly represents the mass ratio of the star to the sun, a unitless mass measurement. When relevant, we refer to the mass of a star with units with an upper-case M. In a similar fashion, l and u refer to the lower and upper mass limits of star formation, respectively.

All nucleosynthetic yields are in fractional units; that is, they quantify the mass fraction of stellar material's initial mass that is processed into a given element and subsequently ejected to the ISM. Nucleosynthetic products that end up locked in stellar remnants should not be taken into account in these models. These values are denoted with a lower-case y with test subscripts and superscripts denoting the element and the enrichment channel.

The metallicity by mass Z refers always to the metallicity by mass:

$$Z \equiv \frac{M_x}{M}$$

Where M_x refers to the mass of some element x and M to the mass of either the interstellar gas or a star.

The logarithmic abundance measurement [X/H] is defined by:

$$[X/H] \equiv \log_{10}\left(\frac{Z_x}{Z_x^\odot}\right)$$

and logarithmic abundance ratios [X/Y]:

$$[X/Y] = [X/H] - [Y/H] = \log_{10}\left(\frac{Z_x}{Z_x^\odot}\right) - \log_{10}\left(\frac{Z_y}{Z_y^\odot}\right)$$

Here and hereafter the symbols $\odot$ and τ refer to the sun and a timescale, respectively.

The term "zone models" refers to both singlezone and multizone models in the general sense.

1.1 Background

1.1.1 Galactic Chemical Evolution

Galactic Chemical Evolution (often referred to as galactic archaeology) studies the connection between a galaxy's evolution and the chemical compositions of its stars. Big Bang Nucleosynthesis produced only hydrogen, helium, and trace amounts of lithium, the three lightest elements on the periodic table. To first order, everything else was produced via nuclear fusion in supernovae and through various channels of stellar evolution, the yields of which are dictated by nuclear physics. The abundances of different nuclei within stars therefore has physical information on the number of nucleosynthetic events and thus the number of stars that came before it. For more theoretical background on galactic archaeology, see sections 1 and 2 and the citations therein of Johnson & Weinberg (2020).

1.1.2 The Singlezone Approximation

The singlezone approximation (also known as the onezone approximation, onezone models, box models, or variations thereof), refers to the assumption of instantaneous diffusion of newly produced metals in interstellar gas. This assumptions mandates that these nuclei be uniformly distributed at all times. By deliberately sacrificing all phase space information, the equations of these models reduce to a system of couple integro-differential equations of mass with time. While these equations only allow analytic solutions under further *mathematical* approximations, they can be easily integrated numerically.

VICE includes features for running numerical simulations of singlezone models in the `singlezone` class. In this documentation, we detail the analytic motivation and numerical approximations implemented in VICE in handling these simulations.

1.1.3 The Multizone Approximation

The multizone approximation refers to the extension of singlezone models to take into account phase space information with any degree of sophistication. By allowing singlezone models to evolve in parallel, and to mix stars and gas amongst them under some prescription, information on the spatial structure of the model galaxy can be added back to the simulation.

VICE includes features for running numerical simulations of multizone models in the `multizone` class. In this documentation, we detail the analytic motivation and numerical approximations implemented in VICE in handling these simulations.

1.2 Implemementation

1.2.1 Motivation

VICE is designed in such a manner that as few assumptions as possible are made by the software itself. In this manner, the power the user has over the parameters of their simulations is maximized. With this motivation, any quantities that may vary are allowed to do so under user-constructed functions in Python. The only assumption VICE's model adopts is physical plausibility.

1.2.2 Numerical Approach

Because VICE is built to handle singlezone and multizone simulations, numerics are not the dominant source of error, but rather in the model itself. The assumption of instantaneous diffusion of newly produced metals introduces an error that which is larger than even modest numerical errors to the equations presented in this documentation.

For this reason, VICE is implemented with a Forward Euler timestep solution, and its errors are not dominated by numerics. Furthermore, quantization of the timesteps allows the quantization of the episodes of star formation with no further assumptions. At several instances in this documentation, this will simplify the equations considerably. Adopting a user-specified timestep size, this also makes it the computationally cheapest solution by not introducing intermediate timesteps. In this manner, VICE is able to achieve a high degree of generality while retaining powerful computing speeds.

1.2.3 Minimization of Dependencies

VICE is implemented in its entirety in ANSI/ISO C, standard library Python, and standard library Cython. With this implementation, VICE is entirely cross platform and independent of the user's version of Anaconda (or lackthereof).

However, VICE is not wrapped for installation in a Windows environment without modifying the installation source code. We recommend users install and run VICE in a linux environment using the Windows Terminal.

1.3 Single Stellar Populations

As discussed in our section on *implementation*, VICE's simulations are implemented with a Forward Euler timestep solution, an approximation made possible by numerics not being the dominant source of error. The quantization of timesteps necessitates the quantization of the episodes of star formation. This allows VICE to model enrichment in both singlezone and multizone models by using summations over a sample of discretized stellar populations.

For this reason, we implement a treatment of two quantities particularly useful in the mass evolution of single stellar populations: the *cumulative return fraction* (CRF) and the *main sequence mass fraction* (MSMF). The *CRF* represents the fraction of a single stellar population's mass that is returned to the interstellar medium as gas. The *MSMF* is the fraction of its mass that is still in the form of main sequence stars. These quantities are of particular use in calculating the rate of mass recycling and the rate of enrichment from asymptotic giant branch stars.

1.3.1 Stellar Lifetimes

In VICE we adopt the following functional form for the lifetime of a star on the main sequence:

$$\tau_\text{MS} = \tau_\odot m^{-\alpha}$$

where $\tau_\odot$ is the sun's main sequence lifetime, α is the power-law index of the mass-lifetime relationship. The constant `SOLAR_LIFETIME` declares $\tau_\odot$ = 10 Gyr, and `MASS_LIFETIME_PLAW_INDEX` delcares α = 3.5. Both constants are declared in `vice/src/ssp.h`.

The scaling of $\tau_\text{MS} \sim m^{-3.5}$ fails for high mass stars ($\gtrsim 8M_\odot$), but these stars have lifetimes that are very short compared to the relevant timescales of galactic chemical evolution (~few' Gyr). This approximation fails for low mass stars as well ($\lesssim 0.5M_\odot$), but these stars have very long lifetimes that are considerably longer than the age of the universe. Because VICE does not support simulations on this long of timescales, this approximation suffices for all timescales of interest.

This is motivated by a conventional power-law relationship between mass and luminosity $L \sim M^{+\beta}$. The lifetime then scales as $\tau \sim M/L \sim M^{1-\beta}$. α = 3.5 corresponds to $L \sim M^{4.5}$ in the mass range of interest.

This equation can be generalized to find the the *total lifetime* of a star of mass m: the time until it produces a remnant by simply amplifying the lifetime by a factor $1 + p_\text{MS}$:

$$\tau_\text{total} = (1 + p_\text{MS})\tau_\odot m^{-\alpha}$$

where p_MS is an adopted lifetime ratio of the post main sequence to main sequence phases of stellar evolution.

By interpreting τ_total as lookback time, we can solve for the mass of remnant producing stars under this model.

$$m_\text{postMS} = \left(\frac{t}{(1 + p_\text{MS})\tau_\odot}\right)^{-1/\alpha}$$

This equation allows the solution of both the *main sequence turnoff mass* and the mass of stars at the end of their post main sequence lifetimes by whether or not p_MS = 0.

Relevant source code:

- `vice/src/ssp.h`
- `vice/src/ssp/mlr.c`

1.3.2 The Cumulative Return Fraction

The cumulative return fraction is defined as the mass fraction of a single stellar population that is returned back to the interstellar medium (ISM) as gas. When dying stars produce their remnants, whatever material that does not end up in the remnant is returned to the ISM. This quantity can be calculated from an initial-final mass relation and an adopted stellar initial mass function (IMF). In short, the cumulative return fraction can be stated mathematically as "ejected material from dead stars in units of total initial amount of material." Its analytic form is therefore given by:

$$r(t) = \int_{m_\text{to}(t)}^u (m - m_\text{rem})\frac{dN}{dm}dm \left[\int_l^u M\frac{dN}{dm}dm\right]^{-1}$$

The current version of VICE employs the initial-final remnant mass relation of Kalirai et al. (2008)[1]:

$$m_\text{rem}(m) = \begin{cases} 1.44 \; (m \geq 8) \\ 0.394 + 0.109m \; (m < 8) \end{cases}$$

For a power-law IMF $dN/dm \sim m^{-\alpha}$, the numerator of $r(t)$ is thus given by:

$$\int_{m_\text{to}(t)}^u (m - m_\text{rem}(m))\frac{dN}{dm}dm = \frac{1}{2-\alpha}m^{2-\alpha}\Bigg|_{m_\text{to}(t)}^u - \frac{1.44}{1-\alpha}m^{1-\alpha}\Bigg|_{m_\text{to}(t)}^u$$

for $m_\text{to}(t) \geq 8$, and

$$\int_{m_\text{to}(t)}^u (m - m_\text{rem}(m))\frac{dN}{dm}dm = \frac{1.44}{1-\alpha}m^{1-\alpha}\Bigg|_8^u + \left[\frac{0.394}{1-\alpha}m^{1-\alpha} + \frac{0.109}{2-\alpha}m^{2-\alpha}\right]_{m_\text{to}(t)}^8$$

for $m_\text{to}(t) < 8$.

This solution is analytic. For piecewise IMFs, this becomes a summation over the relevant mass ranges of the IMF, and each term has the exact same form. The normalization of the IMF is irrelvant here, because the same normalization will appear in the denominator.

The denominator has a simpler analytic form:

$$\int_l^u m\frac{dN}{dm}dm = \frac{1}{2-\alpha}m^{2-\alpha}\Bigg|_l^u$$

Here we plot r as a function of the stellar population's age. Weinberg, Andrews, and Freudenburg (2017)[2] adopted instantaneous recycling, whereby a fraction of the stellar population's mass r_inst is returned *instantaneously* in the interest of an analytic approach to singlezone models. They find that r_inst = 0.4 and r_inst = 0.2 is an adequate approximation for Kroupa and Salpeter IMFs. This reduces the more sophisticated formulation implemented here to:

$$r(t) \approx \begin{cases} r_\text{inst} \; (t = 0) \\ 0 \; (t > 0) \end{cases}$$

In reality, the rate of mass return from a stellar population of mass M_* is given by $\dot{r}M_*$, but in implementation, the quantization of timesteps allows each timestep to represent a single stellar population which will eject mass M_*dr in a time interval dt. For that reason, VICE is implemented with a calculation of $r(t)$ rather than $\dot{r}$.

In simulations, VICE allows users the choice between the time-dependent formulation of $r(t)$ derived here and the instantaneous approximation of Weinberg, Andrews, and Freudenburg (2017) by specifying a preferred value of r_inst, which allows any fraction between 0 and 1.

[1] Kalirai et al. (2008), ApJ, 676, 594

[3] Kroupa (2001), MNRAS, 322, 231

[4] Salpeter (1955), ApJ, 121, 161

[2] Weinberg, Andrews & Freudenburg (2017), ApJ, 837, 183

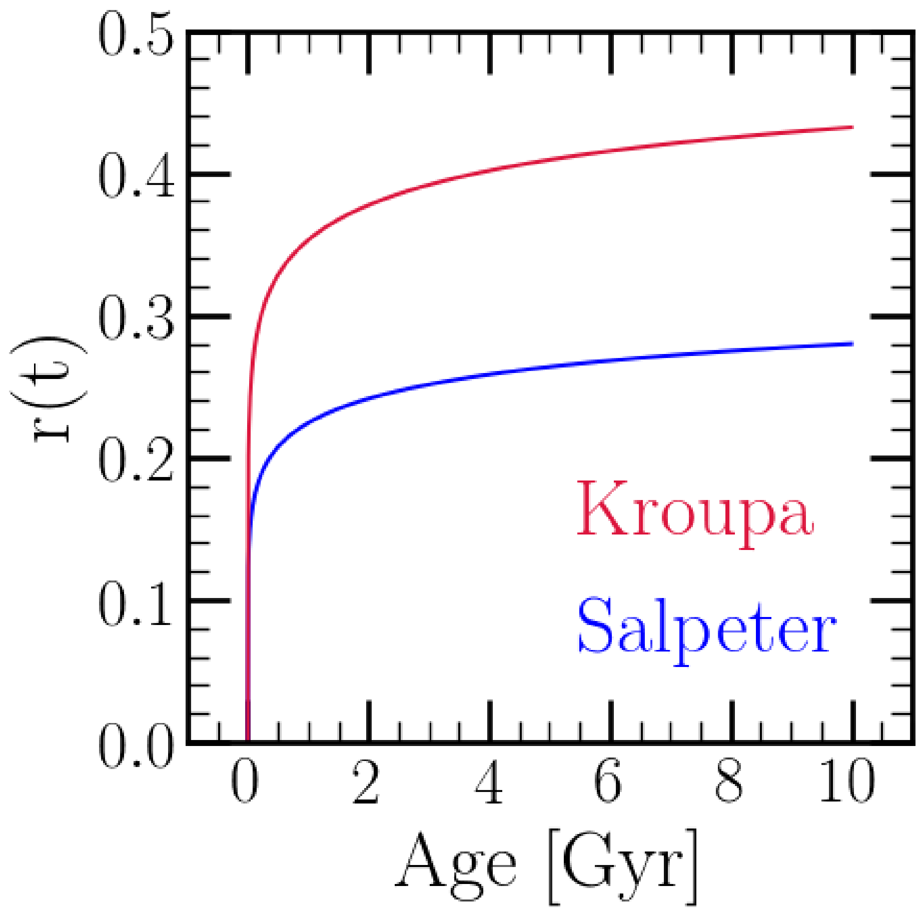

Fig. 1: The cumulative return fraction as a function of age for Kroupa[3] (red) and Salpeter[4] (blue) IMFs. The Kroupa IMF is higher at all nonzero ages because it has fewer low mass stars than Salpeter. In both cases the post main sequence lifetime is assumed to be 10% of the main sequence lifetime (i.e. $p_{\text{MS}} = 0.1$).

In calculations of $r(t)$ with the built-in Kroupa and Salpeter IMFs, the analytic solution is calculated. In the case of a user-customized IMF, VICE solves the equation numerically using quadrature.

Note: The approximation of $h(t) \approx 1 - r(t)$ where h is the *main sequence mass fraction* fails at the $\sim 5 - 10\%$ level. See our discussion of this point *here*.

Relevant source code:

- `vice/src/ssp/crf.c`
- `vice/src/yields/integral.c`

1.3.3 The Main Sequence Mass Fraction

The main sequence mass fraction, as the name suggests, is the fraction of a single stellar population's initial mass that is still in the form of main sequence stars. Because this calculation does not concern evolved stars, neither a model for the post main sequence lifetime nor an initial-final remnant mass relation is needed; it is thus considerably simpler than the *cumulative return fraction*. This quantity is instead specified entirely by the IMF and the mass-lifetime relation.

It's analytic form is given by:

$$h(t) = \int_l^{m_\text{to}(t)} m\frac{dN}{dm}dm \left[\int_l^u m\frac{dN}{dm}dm\right]^{-1}$$

which for a power-law IMF $dN/dm \sim m^{-\alpha}$ becomes

$$h(t) = \left[\frac{1}{2-\alpha}m^{2-\alpha}\Bigg|_l^{m_\text{to}(t)}\right]\left[\frac{1}{2-\alpha}m^{2-\alpha}\Bigg|_l^u\right]^{-1}$$

It may be tempting to cancel the factor of $1/(2-\alpha)$, but more careful consideration must be taken for piece-wise IMFs like Kroupa[1]:

$$h(t) = \left[\sum_i \frac{1}{2-\alpha_i}m^{2-\alpha_i}\right]_l^{m_\text{to}(t)}\left(\left[\sum_i \frac{1}{2-\alpha_i}m^{2-\alpha_i}\right]_l^u\right)^{-1}$$

where the summation is over the relevant mass ranges with different power-law indeces α_i. In the case of kroupa α = 2.3, 1.3, and 0.3 for m > 0.5, 0.08 $\leq m \leq$ 0.5, and m < 0.08, respectively.

Here we plot h as a function of the stellar population's age. By 10 Gyr, $h(t)$ is as low as ~ 0.45 for the Kroupa IMF and ~ 0.65 for the Salpeter IMF. In comparison, the *cumulative return fraction* $r(t) \approx 0.45$ for the Kroupa IMF and ~ 0.28 for the Salpeter IMF. This suggests that the approximation $h(t) \approx 1 - r(t)$ fails at the $\sim 5 - 10\%$ level, depending on the choice of IMF. This suggests that for old stellar populations, a non-negligible portion of the mass is contained in evolved stars and stellar remnants. VICE therefore differentiates between these two quantities in its implementation.

In reality, the rate of the stellar mass evolving off of the main sequence is given by $\dot{h}M_*$ where M_* is the initial mass of the stellar population. However, the quantization of timesteps in VICE allows each timestep to represent a single stellar population which will eject mass M_*dh in a time interval dt. For that reason, VICE is implemented with a calculation of $h(t)$ rather than $\dot{h}$.

In calculations of $h(t)$ with the built-in Kroupa and Salpeter IMFs, the analytic solution is calculated. In the case of a user-customized IMF, VICE solves the equation numerically using quadrature.

Relevant source code:

[1] Kroupa (2001), MNRAS, 322, 231

[2] Salpeter (1955), ApJ, 121, 161

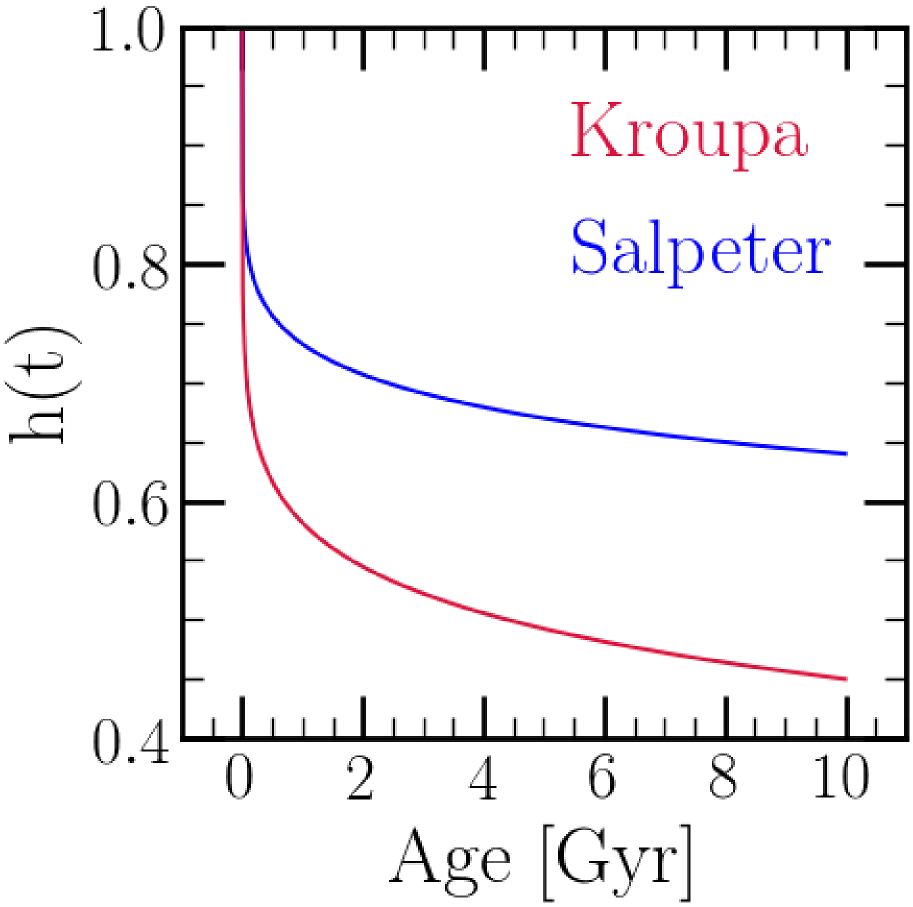

Fig. 2: The main sequence mass fraction as a function of age for Kroupa and Salpeter[2] IMFs. The Kroupa IMF is lower at all nonzero ages because it has fewer low mass stars than Salpeter.

- vice/src/ssp/msmf.c
- vice/src/yields/integral.c

1.4 Enrichment

VICE takes a general approach in modeling nucleosynthesis. All elements are treated equally; there are no special considerations for any element. In this documentation we derive the analytic form of *the enrichment equation* for an arbitrary element x with arbitrary nucleosynthetic yields for arbitrary evolutionary histories. This is an integro-differential equation of the element's mass as a function of time, which VICE solves as an initial-value problem by imposing the boundary condition that its abundance at time zero is given by the primordial abundance from big bang nucleosynthesis. In this version of VICE, helium is the only element for which this value is nonzero.

1.4.1 The Enrichment Equation

The enrichment equation quantifies the rate of change of an element's total mass present in the interstellar medium (ISM). At its core, it is a simple sum of source and sink terms.

$$\dot{M}_x = \dot{M}_x^\text{CC} + \dot{M}_x^\text{Ia} + \dot{M}_x^\text{AGB} - \frac{M_x}{M_g}\left[\dot{M}_\star + \xi_\text{enh}\dot{M}_\text{out}\right] + \dot{M}_x^\text{r} + Z_{x,\text{in}}\dot{M}_\text{in}$$

where M_x is the mass of the element x in the interstellar medium, $\dot{M}_x$ its time-derivative, and M_g the mass of the ISM gas. $\dot{M}_x^\text{CC}$, $\dot{M}_x^\text{Ia}$, and $\dot{M}_x^\text{AGB}$ quantify the rate of production from core-collapse supernovae (CCSNe), type Ia supernovae (SNe Ia), and asymptotic giant branch (AGB) stars, respectively.

We detail each term individually here.

1.4.2 Core Collapse Supernovae

Core collapse supernovae (CCSNe) are the explosions of massive stars ($\gtrsim 8M_\odot$) at the end of their post main sequence lifetimes. Due to the steep nature of the lifetime-stellar mass relationship, these stars have lifetimes that are extremely short compared to the relevant timescales of galactic chemical evolution ($\sim$ few Myr compared to $\sim$ few Gyr). To a good approximation, the lifetimes of these stars can be treated as instantaneous in zone models.

Note: Another motivation for this approximation is that the lifetimes are often significantly shorter than the typical mixing timescales in even modestly sized galaxies. The longest lifetimes of these stars is of order tens of megayears; in comparison, the mixing timescale in the solar annulus of the Milky Way is likely comparable to the dynamical timescale at this distance ($\sim$ 250 Myr, a factor of ten larger). Zone models at their core already assume that these mixing timescales are negligibly short due to the assumption of instantaneous mixing; if CCSN timescales are even shorter, then they can certainly also be modeled as instantaneous.

VICE therefore approximates CCSNe as being simultaneous with the formation of their progenitor stars. This implies a linear relationship between the rate of production of some element x from CCSNe and the star formation rate:

$$\dot{M}_x^\text{CC} = y_x^\text{CC}(Z)\dot{M}_\star$$

where y_x^CC is the *IMF-averaged fractional net yield* of the element x from CCSNe at a metallicity Z: the fraction of the entire stellar population's initial mass that is processed into the element x *and* ejected to the interstellar medium *minus* the amount that the star was born with.

Note: VICE implements recycling of previously produced elements separate from nucleosynthesis, running from the standpoint of *net* rather than *absolute* yields.

In practice, y_x^{CC} is highly uncertain[1]. VICE therefore makes no assumptions about the user's desired form of the yield; this parameter can be assigned either a number to represent a metallicity-independent yield, or a function of the metallicity by mass $Z = M_x/M_g$. VICE includes features which will calculate the value of y_x^{CC} for a given element and metallicity based on the results of supernova nucleosynthesis studies upon request, but requires the user to specify an exact number or function.

Relevant Source Code:

- `vice/src/singlezone/ccsne.c`
- `vice/core/dataframe/_yield_settings.pyx`
- `vice/yields/ccsne/__init__.py`

1.4.3 Type Ia Supernovae

Type Ia supernovae are the thermonuclear detonations of white dwarf stars. Being the remnants of lower-mass stars, white dwarfs are born and explode on timescales longer than the mixing timescales of galaxies. Therefore, the intrinsic time delay is non-negigible.

This requires a model for the SN Ia delay-time distribution (DTD), defined as the rate of SN Ia explosions associated with a single stellar population. Given a DTD R_{Ia} and an age τ, the rate of production of some element x from a single stellar population is given by

$$\dot{M}_x^{\text{Ia}} = y_x^{\text{Ia}}(Z) M_* \frac{R_{\text{Ia}}(\tau)}{\int_0^\infty R_{\text{Ia}}(t)dt}$$

Note: The integral of this equation from $t = 0$ to ∞ must equal the yield times the mass of the stellar population. This necessitates the normalization of the DTD.

where y_x^{Ia} is the *IMF-averaged fractional net yield* of the element x from SNe Ia at metallicity Z: the fraction of the stellar population's initial mass that is processed into the element x *and* ejected to the interstellar medium *minus* the amount that the star was born with.

Note: VICE implements recycling of previously produced elements separate from nucleosynthesis, running from the standpoint of *net* rather than *absolute* yields.

In practice, y_x^{Ia} is highly uncertain[2]. VICE therefore makes no assumptions about the user's desired form of the yield; this parameter can be assigned either a number to represent a metallicity-independent yield or a function of metallicity by mass $Z = M_x/M_g$. VICE includes features which will calculate the value of y_x^{Ia} for a given element and metallicity based on the results of supernova nucleosynthesis studies upon request, but requires the user to specify an exact number or function.

The rate of enrichment from all previous episodes of star formation can be derived by integrating this equation over all ages:

$$\dot{M}_x^{\text{Ia}} = y_x^{\text{Ia}}(Z) \frac{\int_0^t \dot{M}_*(t') R_{\text{Ia}}(t - t')dt'}{\int_0^\infty R_{\text{Ia}}(t')dt'}$$

[1] See Andrews, Weinberg, Schoenrich & Johnson (2017), ApJ, 835, 224 and the citations therein for a detailed analysis of multiple elements.

[2] See Andrews, Weinberg, Schoenrich & Johnson (2017), ApJ, 835, 224 and the citations therein for a detailed analysis of multiple elements.

This can also be expressed as the star formation history up to a time t weighted by the SN Ia rate. VICE approximates this equation as:

$$\dot{M}_x^{\text{Ia}} \approx \frac{\sum_i y_x^{\text{Ia}}(Z_{\text{ISM}}(i\Delta t))\dot{M}_*(i\Delta t)R_{\text{Ia}}(t - i\Delta t)\Delta t}{\sum_i^{T_{\text{Ia}}} R_{\text{Ia}}(i\Delta t)\Delta t}$$

where the sum in the numerator is over all timesteps and in the denominator up to a time T_{Ia} denoting an adopted full length of the SN Ia duty cycle. The constant `RIA_MAX_EVAL_TIME` declares T_{Ia} = 15 Gyr in `vice/src/sneia.h`.

In implementation, VICE normalizes the DTD at the beginning of the simulation. For an age $\tau = t - t'$:

$$R_{\text{Ia}}(\tau) \rightarrow \frac{R_{\text{Ia}}(\tau)}{\int_0^{T_{\text{Ia}}} R_{\text{Ia}}(\tau)d\tau} \approx \frac{R_{\text{Ia}}(t - i\Delta t)}{\sum_i^{T_{\text{Ia}}} R_{\text{Ia}}(i\Delta t)\Delta t} \implies R_{\text{Ia}}(t - t')\Delta t \rightarrow \frac{R_{\text{Ia}}(t - i\Delta t)\Delta t}{\sum_i^{T_{\text{Ia}}} R_{\text{Ia}}(i\Delta t)\Delta t}$$

Inserting the normalized rate into the equation for $\dot{M}_x^{\text{Ia}}$:

$$\dot{M}_x^{\text{Ia}} \approx \sum_i y_x^{\text{Ia}}(Z_{\text{ISM}}(i\Delta t))\dot{M}_*(i\Delta t)R_{\text{Ia}}(t - i\Delta t)$$

VICE implements this normalization of R_{Ia} at the beginning of simulations due to the simplification of this expression introduced in doing so. This reduces the computational expense in calculating this quantity for each element at each timestep.

VICE includes two built-in DTDs, denoting by strings as `plaw` and `exp`. As their names suggest, they are a power-law and an exponential DTD:

- “plaw”: $R_{\text{Ia}} \sim t^{-1.1}$
- “exp”: $R_{\text{Ia}} \sim e^{-t/\tau_{\text{Ia}}}$

Users may also construct their own functional forms of R_{Ia}, which must accept time in Gyr as the only parameter. These functions need not be normalized in any way; VICE normalizes the DTD automatically.

Relevant Source Code:

- `vice/src/sneia.h`
- `vice/src/singlezone/sneia.c`
- `vice/yields/sneia/__init__.py`

1.4.4 Asymptotic Giant Branch Stars

Asymptotic giant branch (AGB) stars are evolved stars that have carbon-oxygen cores surrounded by helium and hydrogen shells. These stars undergo thermal pulsations due to explosive ignition of helium fusion in the shell, typically referred to as helium shell flashes. During these pulses, material from the core is often mixed into the outer layers via convection, a process known as *dredge-up*. This brings heavy nuclei produced in the deeper regions of the star to the envelope, which is then ejected to the interstellar medium (ISM). This is one of the primary sites of s-process nucleosynthesis in the universe.

It may be tempting to model AGB star enrichment as a delay-time distribution (DTD) similar to that adopted for SNe Ia. However, this approach would implicitly adopt the assumption that every element is enriched via AGB stars with the same DTD, or that for a given element, the effective DTD is independent of metallicity. These may be fine assumptions, but it is not adopted in VICE due to the desire for as few assumptions as possible.

Instead, AGB star enrichment in VICE is implemented using the *mass-lifetime relationship for stars* and the *main sequence mass fraction* (MSMF). However, the form of the *MSMF* required here differs in detail from the true *MSMF*. Being evolved stars, the *MSMF* does not consider AGB stars. It is thus not the *MSMF* and the main sequence lifetimes of stars that are of interest, but the mass fraction of both main sequence and evolved stars and the *total* lifetime of stars. The form of $h(t)$ necessary for modeling AGB star enrichment then changes to:

$$h(t) \rightarrow \frac{\int_l^{m_\text{postMS}(t)} m \frac{dN}{dm} dm}{\int_l^u m \frac{dN}{dm} dm}$$

The numerator is evaluated from l to the mass of stars ending their post main sequence lifetime m_postMS rather than the main sequence turnoff mass m_to. As detailed *here* for a stellar population of age τ:

$$m_\text{postMS} = \left(\frac{\tau}{(1 + p_\text{MS})\tau_\odot}\right)^{-1/\alpha}$$

where α is the power-law index on the *mass-lifetime relationship*, $\tau_\odot$ is the main sequence lifetime of the sun, and p_MS is the ratio of a star's post main sequence lifetime to its main sequence lifetime.

From a single stellar population, the rate of ejection of an element x from AGB stars to the ISM is given by:

$$\dot{M}_x^\text{AGB} = -y_x^\text{AGB}(m_\text{postMS}, Z) M_\star \dot{h}$$

where $\dot{h}$ is evaluated at the lookback time to the stellar population's formation[3], $M_\star$ is the initial mass of the stellar population, and y_x^AGB is the *fractional net yield* of x from an AGB star of initial mass m_postMS and metallicity Z: the fraction of a single star's initial mass that is processed into element x *and* ejected to the interstellar medium *minus* the amount that the star was born with.

Note: VICE implements recycling of previously produced elements separate from nucleosynthetic yields, running from the standpoint of *net* rather than *absolute* yields.

For continuous star formation, the enrichment rate can be expressed as this quantity integrated over the star formation history:

$$\dot{M}_x^\text{AGB} = -\int_0^t y_x^\text{AGB}(m_\text{postMS}(t - t'), Z_\text{ISM}(t'))\dot{M}_\star(t')\dot{h}(t - t')dt$$

This expression is approximated numerically as:

$$\dot{M}_x^\text{AGB} \approx \sum_i y_x^\text{AGB}(m_\text{postMS}(t - i\Delta t), Z_\text{ISM}(i\Delta t))\dot{M}_\star(i\Delta t)\left[h((i+1)\Delta t) - h(i\Delta t)\right]$$

where the summation is taken over all previous timesteps. The need to differentiate h with time is eliminated in the numerical approximation by allowing each stellar population to be weighted by Δh between the current timestep and the next, made possible by the quantization of timesteps.

In practice, y_x^AGB is highly uncertain[4]. VICE therefore makes no assumptions about the user's desired form of the yield; this parameter can be assigned either a built-in table published in an AGB star nucleosynthesis study or a function of stellar mass and metallicity constructed by the user.

Relevant source code:

- `vice/src/singlezone/agb.c`
- `vice/core/dataframe/_agb_yield_settings.pyx`
- `vice/yields/agb/__init__.py`

[3] There is a minus sign here because $h(t)$ is a monotonically decreasing function, and thus $\dot{h} < 0$.

[4] See Andrews, Weinberg, Schoenrich & Johnson (2017), ApJ, 835, 224 and the citations therein for a detailed analysis of multiple elements.

1.4.5 Subsequent Terms

The remaining terms in the enrichment equation make simple statements about remaining source and sink terms.

VICE retains the assumption that stars are born at the same metallicity as the ISM from which they form. This motivates the sink term

$$-\left(\frac{M_x}{M_g}\right)\dot{M}_\star$$

where the mass of the element x is depleted at the metallicity of the ISM $Z_x = M_x/M_g$ in proportion with the star formation rate $\dot{M}_\star$.

Many galactic chemical evolution models to date have assumed that outflows from galaxies occur at the same metallicity of the ISM. This would suggest that $\dot{M}_x^\text{out} \approx (M_x/M_g)\dot{M}_\text{out}$. However, recent work in the astronomical literature from both simulations (e.g. Christensen et al. (2018)[5]) and observations (e.g. Chisholm, Trimonti & Leitherer (2018)[6]) suggest that this may not be the case. Therefore, VICE allows outflows to occur at some multiplicative factor ξ_enh above or below the ISM metallicity, which may vary with time. This motivates the sink term

$$-\left(\frac{M_x}{M_g}\right)\xi_\text{enh}\dot{M}_\text{out}$$

Because *VICE works with net rather than absolute yields*, simulations must quantify the rate at which stars return mass to the ISM at their birth metallicity. This is mathematically similar to the rate of total gas recycling, but weighted by the metallicities of the stars recycling. Since stars are assumed to form at the metallicity of the ISM,

$$\dot{M}_x^\text{r} = \int_0^t \dot{M}_\star(t')Z_{x,\text{ISM}}(t')\dot{r}(t - t')dt$$

This is approximated numerically as

$$\dot{M}_x^\text{r} \approx \sum_i \dot{M}_\star(i\Delta t)Z_{x,\text{ISM}}(i\Delta t)\left[r((i+1)\Delta t) - r(i\Delta t)\right]$$

where the summation is taken over all previous timesteps. The need to differentiate r with time is eliminated in the numerical approximation by allowing each stellar population to be weighted by Δr between the current timestep and the next, made possible by the quantization of timesteps. In the event that the user has specified instantaneous recycling:

$$\dot{M}_x^\text{r} = r_\text{inst}\dot{M}_\star Z_{x,\text{ISM}}$$

At any given timestep, there is gas infall onto the simulated galaxy of a given metallicity Z. In most cases this term is negligibly small, but in some interesting cases it may not be (e.g. a major merger event). This necessitates the final term $Z_{x,\text{in}}\dot{M}_\text{in}$.

Relevant source code:

- `vice/src/singlezone/element.c`
- `vice/src/singlezone/ism.c`

1.4.6 Sanity Checks

At all timesteps VICE forces the mass of every element to be non-negative. If the mass is found to be below zero at any given time, it is assumed to not be present in the interstellar medium and is assigned a mass of exactly zero. Absent this, the mass of each element reported by VICE is merely the numerically estimated solution to the enrichment equation.

Relevant source code:

[5] Christensen et al. (2018), ApJ, 867, 142

[6] Chisholm, Trimonti & Leitherer (2018), MNRAS, 481, 1690

- `vice/src/singlezone/element.c`

1.5 Nucleosynthetic Yields

Due to the associated uncertainties[1], VICE takes an agnostic approach to the user's desired nucleosynthetic yields. Rather than adopting the results of a nucleosynthesis study, the user declares their yields outright. VICE includes features which will calculate yields upon request, but requires the user to explicitly tell it what the yield of each element from each enrichment channel should be (although there is a set of defaults).

All yields in VICE are defined as *fractional net yields*. This is the amount of an element that is *produced and ejected* to the interstellar medium *minus* that which was already present, in units of the star or stellar population's initial mass. Previously produced nuclei should not be taken into account, because this is handled via *recycling*. For example, if a stellar population is born with $1M_\odot$ of oxygen total and ejects $1M_\odot$ of oxygen back to the interstellar medium, the yield is zero since there is no net gain.

Yields are also defined for the average star or stellar population. Stochasticity in yields introduced by, e.g., sampling of the stellar initial mass function, should not be taken into account in yield calculations intended for use in VICE.

1.5.1 Core Collapse Supernovae

Because core collapse supernovae (CCSNe) are assumed to occur simultaneously with the formation of their progenitor stars[2], y_x^CC represents the total yield from all CCSNe associated with a single stellar population. Letting m_x denote the net mass of some element x present in the CCSN ejecta, the yield at a given metallicity is defined by:

$$y_x^\text{CC} = \frac{\int_{l_\text{CC}}^u E(m) m_x \frac{dN}{dm} dm}{\int_l^u m \frac{dN}{dm} dm}$$

where the numerator is taken from the minimum mass for a CCSN explosion l_CC to the upper mass limit of star formation u, but the denominator is over the entire mass range of star formation, and dN/dm is the stellar initial mass function (IMF). $E(m)$ denotes the *explodability*: the fraction of stars of mass m which explode as a CCSN. The constant `CC_MIN_STELLAR_MASS` declares $l_\text{CC} = 8M_\odot$ in `vice/src/ccsne.h`. This equation is nothing more than the mathematical statement of "production divided by total initial mass."

In practice, supernova nucleosynthesis studies determine the value of m_x for of order 10 values of m at a given metallicity and rotational velocity. To compute the numerator of this equation, VICE adopts a grid of m_x values from a user-specified nucleosynthesis study, interpolating linearly between values of m on the grid. We clarify that the interpolation is linaer in m, and not $\log m$.

In this version of VICE, users can choose between the following nucleosynthesis studies:

- Limongi & Chieffi (2018), ApJS, 237, 13
- Chieffi & Limongi (2013), ApJ, 764, 21
- Nomoto, Kobayashi & Tominaga (2013), ARA&A, 51, 547
- Chieffi & Limongi (2004), ApJ, 608, 405
- Woosley & Weaver (1995), ApJS, 101, 181

By default, VICE will assume that all stars above $8M_\odot$ explode as a CCSN. Because stellar explodability is an open question in astronomy[3], $E(m)$ can be specified as an arbitrary mathematical function, which must accept stellar mass in $M_\odot$ as the only parameter. Lastly, this can be done with either the built-in Kroupa[4] or Salpeter[5] IMFs, or a function

[1] See Andrews, Weinberg, Schoenrich & Johnson (2017), ApJ, 835, 224 and the citations therein for a detailed analysis of multiple elements.

[2] See the discusion on *CCSN enrichment* for justification of this assumption.

[3] See the discussion in Sukhbold et al. (2016), ApJ, 821, 38 and the citations therein for details.

[4] Kroupa (2001), MNRAS, 231, 322

[5] Salpeter (1955), ApJ, 121, 161

of mass interpreted as a user-constructed IMF.

Note: VICE also forces m_x = 0 at $8M_\odot$, the default value of l_CC, in order to minimize numerical artifacts introduced when extrapolating off of the grid in m to lower stellar masses.

Users can evaluate the solution to this equation by calling the function `vice.yields.ccsne.fractional`, implemented in `vice/yields/ccsne/_yield_integrator.pyx`. This function makes use of numerical quadrature routines written in ANSI/ISO C built into VICE, and is thus not dependent on any publicly available quadrature functions such as those found in `scipy`.

In addition to evaluating the solution to this equation, users may also read in the table of m_x values by calling `vice.yields.ccsne.table`, and may request the full isotopic breakdown. A `dataframe` is returned from this function.

Note: These functions have no impact whatsoever on the chemical enrichment simulations built into VICE. Users declare their own yields for that purpose, while this function merely calculates them.

Relevant Source Code:

- `vice/src/yields/integral.c`
- `vice/yields/ccsne/_yield_integrator.pyx`
- `vice/yields/ccsne/table.py`
- `vice/core/dataframe/_ccsn_yield_table.pyx`

CHAPTER

TWO

USER'S GUIDE

2.1 From the Command Line

Included with VICE is a command line entry which runs simple simulations from a terminal. This feature allows the parameters of a onezone model to be specified as command-line arguments; run `vice --help` from a terminal after installing VICE (from any directory except the source tree). While these command-line capabilities are useful for their ease, VICE is severaly limited in capability when ran from the command-line in comparison to when ran from the Python interpreter.

VICE also includes a command-line entry for automatically accessing the documentation. Simply run `vice-docs` from any directory except the source tree, and the documentation will be opened by the default web browser.

2.2 Package Contents

2.2.1 vice.cumulative_return_fraction

Calculate the cumulative return fraction for a single stellar population at a given age. This quantity represents the fraction of the stellar population's mass that is returned to the interstellar medium as gas at the birst metallicity of the stars.

Signature: vice.cumulative_return_fraction(age, IMF = "kroupa", m_lower = 0.08, postMS = 0.01)

2.2.1.1 Parameters

age [real number] The age of the stellar population in Gyr.

IMF [`str` [case-insensitive] or `<function>` [default]["kroupa"]] The assumed stellar initial mass function (IMF). Strings denote built-in IMFs. Functions must accept only one numerical parameter and will be interpreted as a custom, arbitrary stellar IMF.

Recognized built-in IMFs:

- Kroupa[1]
- Salpeter[2]

Note: Functions do not need to be normalized. VICE will take care of this automatically.

[1] Kroupa (2001), MNRAS, 231, 322
[2] Salpeter (1955), ApJ, 121, 161

m_upper [real number [default][100]] The upper mass limit on star formation in solar masses.

m_lower [real number [default][0.08]] The lower mass limit on star formation in solar masses.

postMS [real number [default][0.1]] The ratio of a star's post main sequence lifetime to its main sequence lifetime.

New in version 1.2.0.

2.2.1.2 Returns

crf [real number] The value of the cumulative return fraction for a stellar population at the specified age under the specified parameters.

2.2.1.3 Notes

Note: VICE operates under the approximation that stars have a mass-luminosity relationship given by:

$$L \sim M^{4.5}$$

leading to a mass-lifetime relation that is also a power law, given by:

$$\tau \sim M/L \sim M^{-3.5}$$

Note: VICE implements the remnant mass model of Kalirai et al. (2008)[3], assuming that stars above 8 $M_\odot$ leave behind remnants of 1.44 $M_\odot$, while stars below 8 $M_\odot$ leave behind remnants of $0.394M_\odot + 0.109M$.

2.2.1.4 Raises

- **TypeError**
  - age is not a real number
  - IMF is neither a string nor a function
  - m_upper is not a real number
  - m_lower is not a real number
  - postMS is not a real number
- **ValueError**
  - age < 0
  - built-in IMF is not recognized
  - m_upper <= 0
  - m_lower <= 0
  - m_lower >= m_upper
  - postMS < 0 or > 1

[3] Kalirai et al. (2008), ApJ, 676, 594

2.2.1.5 Example Code

```
>>> vice.cumulative_return_fraction(1)
0.3560160079575864
>>> vice.cumulative_return_fraction(2)
0.38056657042902253
>>> vice.cumulative_return_fraction(3)
0.394760119115021
```

2.2.2 vice.main_sequence_mass_fraction

Calculate the main sequence mass fraction for a single stellar population at a given age. This quantity represents the fraction of the stellar population's mass that is still in the form of stars on the main sequence.

Signature: vice.main_sequence_mass_fraction(age, IMF = "kroupa", m_upper = 100, m_lower = 0.08)

2.2.2.1 Parameters

age [real number] The age of the stellar population in Gyr.

IMF [`str` [case-insensitive] or `<function>` [default]["kroupa"]] The assumed stellar initial mass function (IMF). Strings denote built-in IMFs. Functions must accept only one numerical parameter and will be interpreted as a custom, arbitrary stellar IMF.

Recognized built-in IMFs:

- Kroupa[1]
- Salpeter[2]

Note: Functions do not need to be normalized. VICE will take care of this automatically.

m_upper [real number [default][100]] The upper mass limit on star formation in solar masses.

m_lower [real number [default][0.08]] The lower mass limit on star formation in solar masses.

2.2.2.2 Returns

msmf [real number] The value of the main sequence mass fraction for a stellar population at the specified age under the specified parameters.

2.2.2.3 Notes

Note: VICE operates under the approximation that stars have a mass-luminosity relationship given by:

$$L \sim M^{4.5}$$

leading to a mass-lifetime relation that is also a power-law, given by:

$$\tau \sim M/L \sim M^{-3.5}$$

[1] Kroupa (2001), MNRAS, 231, 322

[2] Salpeter (1955), ApJ, 121, 161

2.2.2.4 Raises

- **TypeError**
  - age is not a real number
  - IMF is neither a string nor a function
  - m_upper is not a real number
  - m_lower is not a real number
  - postMS is not a real number
- **ValueError**
  - age < 0
  - built-in IMF is not recognized
  - m_upper <= 0
  - m_lower <= 0
  - m_lower >= m_upper

2.2.2.5 Example Code

```
>>> vice.main_sequence_mass_fraction(1)
0.5815004968281556
>>> vice.main_sequence_mass_fraction(2)
0.5445877675278488
>>> vice.main_sequence_mass_fraction(3)
0.5219564300200146
```

2.2.3 vice.single_stellar_population

Simulate the nucleosynthesis of a given element from a single star cluster of given mass and metallicity. This does not take into account galactic evolution - whether or not it is depleted from inflows or ejected in winds is not considered. Only the mass of the given element produced by the star cluster is calculated.

Signature: vice.single_stellar_population(element, mstar = 1.0e+06, Z = 0.014, time = 10, dt = 0.01, m_upper = 100, m_lower = 0.08, postMS = 0.1, IMF = “kroupa”, RIa = “plaw”, delay = 0.15)

2.2.3.1 Parameters

element [`str` [case-insensitive]] The symbol of the element to simulate the enrichment for.

mstar [real number [default][1.0e+06]] The birth mass of the star cluster in solar masses.

Z [real number [default][0.014]] The metallicity by mass of the stars in the cluster.

time [real number [default][10]] The amount of time in Gyr to run the simulation for

dt [real number [default][0.01]] The size of each timestep in Gyr

m_upper [real number [default][100]] The upper mass limit on star formation in solar masses.

m_lower [real number [default][0.08]] The lower mass limit on star formation in solar masses.

postMS [real number [default][0.1]] The ratio of a star's post main sequence lifetime to its main sequence lifetime.

New in version 1.2.0.

IMF [`str` [case-insensitive] or `<function>` [default]["kroupa"]] The stellar initial mass function (IMF) to assume. Strings denote built-in IMFs. Functions must accept only one numerical parameter and will be interpreted as a custom, arbitrary stellar IMF.

Recognized built-in IMFs:

- Kroupa[1]
- Salpeter[2]

Note: Functions do not need to be normalized. VICE will take care of this automatically.

RIa [`str` [case-insensitive] or `<function>` [default]["plaw"]] The delay-time distribution for type Ia supernovae to adopt. Strings denote built-in distributions. Functions must accept only one numerical parameter and will be interpreted as a custom, arbitrary delay-time distribution.

Recognized built-in distributions:

- "plaw": $R_\text{Ia} \sim t^{-1.1}$
- "exp": $R_\text{Ia} \sim e^{-t/1.5\text{ Gyr}}$

Note: Functions do not need to return 0 at times smaller than the SN Ia minimum delay time.

Note: Functions do not need to be normalized. VICE will take care of this automatically.

delay [real number [default][0.15]] The minimum delay time following the formation of a single stellar population before the onset of type Ia supernovae in Gyr.

agb_model [string [case-insensitive] or None [default][None]] **[DEPRECATED]**

A keyword denoting which table of nucleosynthetic yields from AGB stars to adopt.

Recognized Keywords:

- "cristallo11"[3]
- "karakas10"[4]

Deprecated since version 1.2.0: Users should instead modify their AGB star yield settings through `vice.yields.agb.settings`. Users may specify either a built-in study or a function of stellar mass and metallicity.

2.2.3.2 Returns

mass [list] The net mass of the element in solar mass produced by the star cluster at each timestep.

times [list] The times in Gyr corresponding to each mass yield.

[1] Kroupa (2001), MNRAS, 231, 322

[2] Salpeter (1955), ApJ, 121, 161

[3] Cristallo et al. (2011), ApJS, 197, 17

[4] Karakas (2010), MNRAS, 403, 1413

2.2.3.3 Raises

- **ValueError**
  - The element is not built into VICE.
  - mstar < 0
  - Z < 0
  - **time < 0 or time > 15 [VICE does not simulate enrichment on** timescales significantly longer than the age of the universe]
  - dt < 0
  - m_upper < 0
  - m_lower < 0
  - m_lower > m_upper
  - postMS < 0 or > 1
  - built-in IMF is not recognized
  - delay < 0
  - agb_model is not built into VICE
- **LookupError**
  - **agb_model == "karakas10" and the atomic number of the element is** larger than 29. The Karakas (2010), MNRAS, 403, 1413 study did not report yields for elements heavier than nickel.
- **ArithmeticError**
  - **A functional RIa evaluated to a negative value, inf, or NaN at any** given timestep.
- **IOError [Only occurs if VICE's file structure has been tampered with]**
  - The AGB yield file is not found.

2.2.3.4 Example Code

```
>>> mass, times = vice.single_stellar_population("sr", Z = 0.008)
>>> mass[-1]
    0.04808964406448721
>>> mass, times = vice.single_stellar_population("fe")
>>> mass[-1]
    2679.816051685778
```

2.2.4 vice.imf.kroupa

The (unnormalized) Kroupa (2001)[1] stellar initial mass function (IMF).

Signature: vice.imf.kroupa(mass)

[1] Kroupa (2001), MNRAS, 322, 231

2.2.4.1 Parameters

mass [real number] The stellar mass in solar masses.

2.2.4.2 Returns

dndm [real number] The unnormalized value of the Kroupa IMF at that stellar mass, defined by:

$$\frac{dN}{dm} \sim m^{-\alpha}$$

where α = 2.3, 1.3, and 0.3 for m > 0.5, 0.08 $\leq m \leq$ 0.5, and m < 0.08, respectively.

2.2.4.3 Raises

- **TypeError**
  - mass is not a real number
- **ValueError**
  - mass is non-positive

2.2.4.4 Example Code

```
>>> vice.imf.kroupa(1)
0.04
>>> vice.imf.kroupa(0.5)
0.1969831061351866
>>> vice.imf.kroupa(2)
0.008122523963562356
```

2.2.5 vice.imf.salpeter

The (unnormalized) Salpeter (1955)[1] stellar initial mass function (IMF).

Signature: vice.imf.salpeter(mass)

2.2.5.1 Parameters

mass [real number] The stellar mass in solar masses.

2.2.5.2 Returns

dndm [real number] The unnormalized value of the Salpeter IMF at that stellar mass, defined by:

$$\frac{dN}{dm} \sim m^{-\alpha}$$

where α = 2.35 always.

[1] Salpeter (1955), ApJ, 121, 161

2.2.5.3 Raises

- **TypeError**
  - mass is not a real number
- **ValueError**
  - mass is non-positive

2.2.5.4 Example Code

```
>>> vice.imf.salpeter(1)
1.0
>>> vice.imf.salpeter(0.5)
5.098242509277049
>>> vice.imf.salpeter(2)
0.19614602447418766
```

2.2.6 vice.singlezone

An object designed to run simulations of chemical enrichment under the single-zone approximation for user-specified parameters. The parameters of the simulation are implemented as attributes of this class.

Signature: Signature: vice.singlezone(**kwargs)

2.2.6.1 Parameters

kwargs [varying types] Every attribute of this class can be assigned via a keyword argument.

2.2.6.2 Attributes

name [`str` [default][“onezonemodel”]] The name of the simulation. Output will be stored in a directory under this name.

func [`<function>` [default][vice._globals._DEFAULT_FUNC_]] A function of time describing some evolutionary parameter. Physical interpretation set by the attribute `mode`.

mode [`str` [default][“ifr”]] The interpretation of the attribute `func`. Either “ifr” for infall rate, “sfr” for star formation rate, or “gas” for the mass of gas.

verbose [`bool` [default][False]] Whether or not to print to the console as the simulation runs.

elements [tuple [default][(“fe”, “sr”, “o”)]] A tuple of strings holding the symbols of the elements to be simulated.

IMF [`str` [case-insensitive] or `<function>` [default][“kroupa”]] The stellar initial mass function (IMF) to adopt. Either a string denoting a built-in IMF or a function containing a user-constructed IMF.

Recognized built-in IMFs:

- “kroupa”[1]
- “salpeter”[2]

[1] Kroupa (2001), MNRAS, 231, 322

[2] Salpeter (1955), ApJ, 121, 161

eta [real number [default][2.5]] The mass-loading parameter: the ratio of outflow to star formation rates. This changes when the attribute `smoothing` is nonzero.

enhancement [real number or `<function>` [default][1]] The ratio of outflow to ISM metallicities. Numbers are interpreted as constants. Functions must accept time in Gyr as a parameter.

Zin [real number, `<function>`, or `dataframe` [default][0]] The infall metallicity, which can be a constant, time-vary, or have element-by-element specifications.

recycling [`str` [case-insensitive] or real number] [default : "continuous"] Either the string "continuous" or a real number between 0 and 1. Denotes the prescription for recycling of previously produced heavy nuclei.

bins [array-like [default][[-3.0, -2.95, -2.9, ... , 0.9, 0.95, 1.0]]] The binspace within which to sort the normalized stellar metallicity distribution function in each [X/H] and [X/Y] abundance ratio measurement.

delay [real number [default][0.15]] The minimum delay time in Gyr before the onset of type Ia supernovae associated with a single stellar population

RIa [`str` [case-insensitive] or `<function>` [default]["plaw"]] The SN Ia delay-time distribution (DTD) to adopt. Strings denote built-in DTDs and functions must accept time in Gyr as a parameter.

Mg0 [real number [default][6.0e+09]] The initial gas supply of the galaxy in solar masses. This is only relevant when the simulation is ran in infall mode (i.e. mode == "ifr").

smoothing [real number [default][0]] The outflow smoothing timescale in Gyr.[3]

tau_ia [real number [default][1.5]] The e-folding timescale of type Ia supernovae in gyr when the attribute `RIa` == "exp".

tau_star [real number or `<function>` [default][2.0]] The star formation rate per unit gas mass in the galaxy in Gyr. This can be either a number which will be treated as a constant, or a function of time in Gyr. This changes when the attribute `schmidt` == True.

dt [real number [default][0.01]] The timestep size in Gyr.

schmidt [`bool` [default][False]] A boolean describing whether or not to implement a gas-dependent star formation efficiency.

schmidt_index [real number [default][0.5]] The power-law index of gas-dependent star formation efficiency.

MgSchmidt [real umber [default][6.0e+09]] The normalization of the gas-supply when the attribute `schmidt` == True.

m_upper [real number [default][100]] The upper mass limit on star formation in solar masses

m_lower [real number [default][0.08]] The lower mass limit on star formation in solar masses

postMS [real number [default][0.1]] The lifetime ratio of the post main sequence to main sequence phases of stellar evolution.

Z_solar [real number [default][0.014]] The adopted metallicity by mass of the sun.

agb_model [`str` [case-insensitive] [default][None]] **[DEPRECATED]**

A keyword denoting which table of nucleosynthetic yields from AGB stars to adopt.

Recognized Keywords:

- "cristallo11"[4]
- "karakas10"[5]

[3] Johnson & Weinberg (2020), arxiv:1911.02598

[4] Cristallo et al. (2011), ApJS, 197, 17

[5] Karakas (2010), MNRAS, 403, 1413

Deprecated since version 1.2: Users should instead modify their AGB star yield settings through `vice.yields.agb.settings`. Users may specify either a built-in study or a function of stellar mass and metallicity.

2.2.6.3 Functions

run [[instancemethod]] Run the simulation

from_output [[classmethod]] Obtain a singlezone object with the parameters of the one that produced an output.

2.2.6.4 Example Code

```
>>> import vice
>>> sz = vice.singlezone()
>>> sz
vice.singlezone{
        name -----------> onezonemodel
        func -----------> <function _DEFAULT_FUNC_ at 0x112180ae8>
        mode -----------> ifr
        verbose --------> False
        elements -------> ('fe', 'sr', 'o')
        IMF ------------> kroupa
        eta ------------> 2.5
        enhancement ----> 1.0
        entrainment ----> <entrainment settings>
        Zin ------------> 0.0
        recycling ------> continuous
        delay ----------> 0.15
        RIa ------------> plaw
        Mg0 ------------> 6000000000.0
        smoothing ------> 0.0
        tau_ia ---------> 1.5
        tau_star -------> 2.0
        schmidt --------> False
        schmidt_index --> 0.5
        MgSchmidt ------> 6000000000.0
        dt -------------> 0.01
        m_upper --------> 100.0
        m_lower --------> 0.08
        postMS ---------> 0.1
        Z_solar --------> 0.014
        bins -----------> [-3, -2.95, -2.9, ... , 0.9, 0.95, 1]
}
```

2.2.7 vice.singlezone.from_output

Obtain an instance of the `singlezone` class given either the path to an output or an output itself.

Signature: vice.singlezone.from_output(arg)

New in version 1.1.0.

2.2.7.1 Parameters

arg [`str` or `output`] The full or relative path to the output directory. Alternatively, an output object.

2.2.7.2 Returns

sz [singlezone] A singlezone object with the same parameters as the one which produced the output.

2.2.7.3 Raises

- **TypeError**
  - arg is neither an output object nor a string
- **IOError**
  - output is not found, or is missing files

2.2.7.4 Notes

Note: If arg is either a multizone output or a multioutput object, a multizone object will be returned.

Note: In versions before 1.1.0, this function had the call signature vice.mirror (now deprecated).

Note: This function serving as the reader, the writer is the vice.core.singlezone._singlezone.c_singlezone.pickle function, implemented in Cython.

2.2.7.5 Example Code

```
>>> import numpy as np
>>> import vice
>>> vice.singlezone(name = "example").run(np.linspace(0, 10, 1001))
>>> sz = vice.singlezone.from_output("example")
vice.singlezone{
        name -----------> example
        func -----------> <function _DEFAULT_FUNC_ at 0x10d0c8e18>
        mode -----------> ifr
        verbose --------> False
        elements -------> ('fe', 'sr', 'o')
        IMF ------------> kroupa
        eta ------------> 2.5
        enhancement ----> 1.0
        entrainment ----> <entrainment settings>
        Zin ------------> 0.0
        recycling ------> continuous
        delay ----------> 0.15
        RIa ------------> plaw
        Mg0 ------------> 6000000000.0
        smoothing ------> 0.0
        tau_ia ---------> 1.5
        tau_star -------> 2.0
        schmidt --------> False
        schmidt_index --> 0.5
```

(continues on next page)

(continued from previous page)

```
        MgSchmidt ------> 6000000000.0
        dt -------------> 0.01
        m_upper --------> 100.0
        m_lower --------> 0.08
        postMS ---------> 0.1
        Z_solar --------> 0.014
        bins -----------> [-3, -2.95, -2.9, ... , 0.9, 0.95, 1]
}
```

2.2.8 vice.singlezone.run

Run the simulation.

Signature: vice.singlezone.run(output_times, capture = False, overwrite = False)

2.2.8.1 Parameters

output_times [array-like [elements are real numbers]] The times in Gyr at which VICE should record output from the simulation. These need not be sorted from least to greatest.

capture [`bool` [default][False]] If True, an output object containing the results of the simulation will be returned.

overwrite [`bool` [default][False]] If True, will force overwrite any files with the same name as the simulation output files.

2.2.8.2 Returns

out [`output` [only returned if `capture` == True]] An output object produced from this simulation's output.

2.2.8.3 Raises

- **TypeError**
  - Any functional attribute evaluates to a non-numerical value.
- **ValueError**
  - Any element of output_times is negative.
  - An inflow metallicity evaluates to a negative value.
- **ArithmeticError**
  - Any functional attribute evaluates to NaN or inf.
- **UserWarning**
  - **Any yield settings or class attributes are callable and the** user does not have dill installed.
- **ScienceWarning**
  - **Any element tracked by the simulation is enriched in signifcant** part by r-process nucleosynthesis.
  - **Any element tracked by the simulation has a weakly constrained** solar abundance measurement.

2.2.8.4 Notes

Note: Calling this function only causes VICE to produce the output files. The `output` class handles the reading and storing of the simulation results.

Note: Saving functional attributes with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

Note: When `overwrite` == False, and there are files under the same name as the output produced, this acts as a halting function. VICE will wait for the user's approval to overwrite existing files in this case. If user's are running multiple simulations and need their integrations not to stall, they must specify `overwrite` = True.

2.2.8.5 Example Code

```
>>> import numpy as np
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> outtimes = np.linspace(0, 10, 1001)
>>> sz.run(outtimes)
```

2.2.9 vice.singlezone.name

Type : `str`

Default : "onezonemodel"

The name of the simulation. The output will be stored in a directory under this name with the extension ".vice". This can also be of the form `./path/to/directory/name` and the output will be stored there.

Tip: Users need not interact with any of the output files. The output object is designed to read in all of the results automatically.

Tip: By forcing a ".vice" extension on the output file, users can run "<command> *.vice" in a terminal to run commands over all VICE outputs in a given directory.

Note: The outputs of this class include the full time evolution of the interstellar abundances, the resulting stellar metallicity distribution, and pickled objects that allow a singlezone object to construct itself from the output. By separating the output into a handful of files, the full time evolution data and the resulting stellar metallicity distribution can be stored in pure ascii text files. This allows users to analyze their simulations in languages other than Python with ease. Most of the relevant information is stored in the history.out and mdf.out files within the output directory.

2.2.9.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.name = "another_name"
```

2.2.10 vice.singlezone.func

Type : `<function>`

Default : vice._globals._DEFAULT_FUNC_

A callable object which must accept time in Gyr as the only parameter. The value returned by this function will represent either the gas infall history in $M_\odot$ yr^{-1} (`mode` == "ifr"), the star formation history in $M_\odot$ yr^{-1} (`mode` == "sfr"), or the ISM gas supply in $M_\odot$ (`mode` == "gas).

Note: The default function returns the value of 9.1 always. With a default `mode` of "ifr", this corresponds to an infall rate of 9.1 $M_\odot$ yr^{-1} at all times.

Note: Saving this functional attribute with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

Note: This attribute will always be expected to accept time in Gyr as the only parameter. However, infall and star formation rates will be interpreted as having units of $M_\odot$ yr^{-1} according to convention.

See also:

`vice.singlezone.mode`

2.2.10.1 Example Code

```
>>> import math as m
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> def f(t):
        if t <= 1:
                return 10
        else:
                return 10 * m.exp(-(t - 1) / 3)
>>> sz.func = f
>>> sz.func = lambda t: 10. * m.exp(-t / 3)
```

2.2.11 vice.singlezone.mode

Type : `str` [case-insensitive]

Default : "ifr"

The interpretation of the attribute `func`.

- **mode = "ifr"** [The value returned from the attribute `func`] represents the rate of gas infall into the interstellar medium in $M_\odot\ yr^{-1}$.
- **mode = "sfr"** [The value returned from the attribute `func`] represents the star formation rate of the galaxy in $M_\odot\ yr^{-1}$.
- **mode = "gas"** [The value returned from the attribute `func`] represents the mass of the ISM gas in $M_\odot$.

Note: The attribute `func` will always be expected to accept time in Gyr as the only parameter. However, infall and star formation rates will be interpreted as having units of $M_\odot\ yr^{-1}$ according to convention.

See also:

`vice.singlezone.func`

2.2.11.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.mode = "sfr"
>>> sz.mode = "gas"
```

2.2.12 vice.singlezone.verbose

Type : `bool`

Default : `False`

If True, the simulation will print to the console as it evolves.

2.2.12.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.verbose = True
```

2.2.13 vice.singlezone.elements

Type : tuple [elements of type str [case-insensitive]]

Default : ("fe", "sr", "o")

The symbols for the elements to track the enrichment for (case-insensitive). The more elements that are tracked, the longer the simulation will take, but the better calibrated is the total metallicity of the ISM in handling metallicity-dependent yields.

Tip: The order in which the elements appear in this tuple will dictate the abundance ratios that are quoted in the final stellar metallicity distribution function. That is, if element X appears before element Y, then VICE will determine the MDF in $dN/d[Y/X]$ as opposed to $dN/d[X/Y]$. The elements that users intend to use as "reference elements" should come earliest in this list.

Note: All versions of VICE support the simulation of all 76 astrophysically produced elements between carbon ("c") and bismuth ("bi"). Versions >= 1.2.0 also support helium ("he").

Note: Some of the heaviest elements that VICE recognizes have statistically significant enrichment from r-process nucleosynthesis[1]. Simulations of these elements with realistic parameters and realistic nucleosynthetic yields will underpredict the absolute abundances of these elements. However, if these nuclei are assumed to be produced promptly following the formation of a single stellar population, the yield can be added to the yield from core collapse supernovae[2].

2.2.13.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.elements
("fe", "sr", "o")
>>> sz.elements = ["mg", "fe", "c", "n", "o"]
>>> sz.elements
("mg", "fe", "c", "n", "o")
```

2.2.14 vice.singlezone.IMF

Type : `str` [case-insensitive] or `<function>`

Default : "kroupa"

New in version 1.2: In versions >= 1.2.0, users may construct a function of mass to describe the IMF.

The assumed stellar initial mass function (IMF). If assigned a string, VICE will adopt a built-in IMF. Functions must accept stellar mass as the only parameter and is expected to return the value of the IMF at that mass.

Built-in IMFs:

- "kroupa"[1]
- "salpeter"[2]

Note: VICE has analytic soluations to the *cumulative return fraction* and the *main sequence mass fraction* for built-in IMFs. If assigned a function, VICE will calculate these quantities numerically, increasing the required integration time.

2.2.14.1 Example Code

[1] Johnson (2019), Science, 363, 474

[2] Johnson & Weinberg (2020), arxiv:1911.02598

[1] Kroupa (2001), MNRAS, 322, 231

[2] Salpeter (1955), ApJ, 121, 161

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.IMF = "kroupa"
>>> def f(m):
        if m < 0.5:
                return m**-1.2
        else:
                return m**-2.2
>>> sz.IMF = f
```

2.2.15 vice.singlezone.eta

Type : real number or `<function>`

Default : 2.5

The mass loading factor, defined as the ratio of the mass outflow rate to the star formation rate.

$$\eta \equiv \frac{\dot{M}_\text{out}}{\dot{M}_*}$$

Note: If the attribute `smoothing` is nonzero, this relationship generalizes to

$$\dot{M}_\text{out} = \eta(t)\langle\dot{M}_*\rangle_{\tau_\text{s}} = \begin{cases} \frac{\eta(t)}{t}\int_0^t \dot{M}_*(t')dt' \ (t < \tau_\text{s}) \\ \frac{\eta(t)}{\tau_\text{s}}\int_{t-\tau_\text{s}}^t \dot{M}_*(t')dt' \ (t \geq \tau_\text{s}) \end{cases}$$

where τ_s is the value of the attribute, the outflow smoothing time.

Note also that the time-average is over the star formation rate only, and not the mass-loading factor.

Note: Saving this functional attribute with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

2.2.15.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.eta = 2
>>> def f(t):
        if t <= 2:
                return 5
        else:
                return 1
>>> sz.eta = f
```

2.2.16 vice.singlezone.enhancement

Type : real number or `<function>`

Default : 1.0

The ratio of the outflow to ISM metallicities. Real numbers will be taken as constant. Functions must accept time in Gyr as the only parameter. This will apply to all elements tracked by the simulation.

Note: Saving this functional attribute with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

See also:

`vice.singlezone.eta` `vice.singlezone.smooting`

2.2.16.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.enhancement = 3
>>> def f(t):
        if t <= 1:
                return 5
        else:
                return 1
>>> sz.enhancement = f
```

2.2.17 vice.singlezone.entrainment

Type : `<entrainment object>`

Default : all elements from all enrichment channels assigned a value of 1.

Each element from each enrichment channel assigned a value of 1. These values denote the mass fraction of nucleosynthetic yields that are retained by the interstellar medium, the remainder of which is added directly to outflows. This must always be numerical values between 0 and 1.

2.2.17.1 Attributes

agb [`dataframe`] The entrainment fraction of each element from AGB stars

ccsne [`dataframe`] The entrainment fraction of each element from CCSNe

sneia [`dataframe`] The entrainment fraction of each element fron SNe Ia

See also:

`vice.dataframe`

2.2.17.2 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> # set the entrainment of CCSN ejecta to 80 percent
>>> for i in sz.elements:
        sz.entrainment.ccsne[i] = 0.8
>>> # set the entrainment of SN Ia ejecta to 90 percent
>>> for i in sz.elements:
        sz.entrainment.sneia[i] = 0.9
```

2.2.18 vice.singlezone.Zin

Type : real number, `<function>`, or `dataframe`

Default : 0.0

The metallicity of gas inflow. Numbers and functions apply to all elements tracked by the simulation. Functions must accept time in Gyr as the only parameter. A dictionary or a `dataframe` can also be passed, allowing real numbers and functions to be assigned on an element-by-element basis.

Tip: The easiest way to switch this attribute to a dataframe is by passing an empty python dictionary `{}`.

Note: Dictionaries will be automatically converted into a `dataframe`.

Note: Saving functional attributes with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

2.2.18.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.Zin = 0.001
>>> def f(t):
        return 0.001 * (t / 5)
>>> sz.Zin = lambda t: 0.001 * (t / 5)
>>> sz.Zin = {}
>>> sz.Zin
vice.dataframe{
        sr -------------> 0.0
        fe -------------> 0.0
        o --------------> 0.0
}
>>> sz.Zin["o"] = 0.001
>>> sz.Zin["fe"] = lambda t: 1.0e-04 * (t / 5)
>>> sz.Zin
vice.dataframe{
        sr -------------> 0.0
        fe -------------> <function main.<__lambda__>(t)>
        o --------------> 0.001
}
```

2.2.19 vice.singlezone.recycling

Type : real number or `str` [case-insensitive]

Default : "continuous"

The *cumulative return fraction* $r(t)$. This is the mass fraction of a single stellar population returned to the interstellar medium as gas at the birth metallicity of the stars.

The only allowed string is “continuous” [case-insensitive]. In this case VICE will implement time-dependent recycling from each episode of star formation via a treatment of the stellar initial mass function and the initial-final remnant mass model of Kalirai at al. (2008)[1].

Numbers must be between 0 and 1 (inclusive), and will be interpreted as the instantaneous recycling fraction: the fraction of a stellar population’s mass that is returned to the interstellar medium immediately following its formation.

Note: In the case of instantaneous recycling, it is recommened that users adopt r = 0.4 with the Kroupa[2] IMF and r = 0.2 with the Salpeter[3] IMF based on the findings of Weinberg, Andrews & Freudenburg (2017)[4].

2.2.19.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example", IMF = "kroupa")
>>> sz.recycling = 0.4
>>> sz.IMF = "salpeter"
>>> sz.recycling = 0.2
>>> sz.recycling = "continuous"
```

2.2.20 vice.singlezone.bins

Type : array-like [elements must be real numbers]

Default : [-3, -2.95, -2.9, . . . , 0.9, 0.95, 1.0]

The bins in each [X/H] abundance and [X/Y] abundance ratio to sort the normalized stellar metallicity distribution function into. By default, VICE sorts everything into 0.05-dex bins between [X/H] and [X/Y] = -3 and +1.

Note: The metallicity distributions reported by VICE are normalized to probability distribution functions (i.e. the integral over all bins is equal to 1).

2.2.20.1 Example Code

```
>>> import numpy as np
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> # 400 bins between 0 and 1
>>> sz.bins = np.linspace(-3, 1, 401)
>>> # 800 bins between -2 and +2
>>> sz.bins = np.linspace(-2, 2, 801)
```

2.2.21 vice.singlezone.delay

Type : real number

[1] Kalirai et al. (2008), ApJ, 676, 594

[2] Kroupa (2001), MNRAS, 231, 322

[3] Salpeter (1955), ApJ, 131, 161

[4] Weinberg, Andrews & Freudenburg (2017), ApJ, 837, 183

Default : 0.15

The minimum delay time in Gyr before the onset of type Ia supernovae associated with a single stellar population. Default value is adopted from Weinberg, Andrews & Freudenburg (2017)[1].

See also:

`vice.singlezone.RIa`

2.2.22 vice.singlezone.RIa

Type : `<function>` or `str` [case-insensitive]

Default : "plaw"

The delay-time distribution (DTD) for typa Ia supernovae to adopt. If type `str`, VICE will use a built-in DTD:

- "exp" : $R_{\text{Ia}} \sim e^{-t}$
- "plaw" : $R_{\text{Ia}} \sim t^{-1.1}$

When using the exponential DTD, the e-folding timescale is set by the attribute `tau_ia`.

Functions must accept time in Gyr as the only parameter.

Tip: A custom DTD does not need to be normalized by the user. VICE will take care of this automatically.

Note: Saving functional attributes with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

2.2.22.1 Example Code

```
>>> import math as m
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.RIa = "exp"
>>> def f(t):
        if t < 0.2:
                return 1
        else:
                return m.exp(-(t - 0.2) / 1.4)
>>> sz.RIa = f
```

2.2.23 vice.singlezone.Mg0

Type : real number

Default : 6.0e+09

The mass of the ISM gas at time = 0 in $M_{\odot}$.

[1] Weinberg, Andrews & Freudenburg (2017), ApJ, 837, 183

Note: This parameter only matters when the simulation is ran in infall mode (i.e. `mode` == "ifr"). In gas mode, `func(0)` specifies the initla gas supply, and in star formation mode, it is `func(0) * tau_star(0)` (modulo the prefactors imposed by gas-dependent star formation efficiency, if applicable).

2.2.23.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.Mg0 = 5.0e+09
>>> sz.Mg0 = 0.
```

2.2.24 vice.singlezone.smoothing

Type : real number

Default : 0.0

The outflow smoothing in Gyr (Johnson & Weinberg 2020[1]). This is the timescale on which the star formation rate is time-averaged before determining the outflow rate via the mass loading factor (attribute `eta`). For an outflow rate $\dot{M}_\text{out}$ and a star formation rate $\dot{M}_*$ with a smoothing time τ_s:

$$\dot{M}_\text{out} = \eta(t)\langle\dot{M}_*\rangle_{\tau_\text{s}}$$

The traditional relationship of $\dot{M}_\text{out} = \eta\dot{M}_*$ is recovered when the user specifies a smoothing time that is smaller than the timestep size.

Note: While this parameter time-averages the star formation rate, it does NOT time-average the mass-loading factor.

2.2.24.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.smoothing = 0.0
>>> sz.smoothing = 0.5
>>> sz.smoothing = 1.0
```

2.2.25 vice.singlezone.tau_ia

Type : real number

Default : 1.5

The e-folding timescale in Gyr of an exponentially decaying delay-time distribution in type Ia supernovae.

Note: Because this is an e-folding timescale, it only matter when the attribute `RIa` == "exp".

[1] Johnson & Weinberg (2020), arxiv:1911.02598

See also:

`vice.singlezone.RIa`

2.2.25.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example", RIa = "exp")
>>> sz.tau_ia = 1.0
>>> sz.tau_ia = 1.5
>>> sz.tau_ia = 2.0
```

2.2.26 vice.singlezone.tau_star

Type : real number or `<function>`

Default : 2.0

The star formation rate per unit gas supply in Gyr, defined by

$$\tau_* \equiv M_\text{g}/\dot{M}_*$$

where M_g is the ISM gas mass and $\dot{M}_*$ is the star formation rate. Numbers will be interpreted as a constant value. Functions must accept time in Gyr as the only parameter.

Tip: In infall and gas modes, this parameter can be set to infinity to forcibly shut off star formation.

Note: When the attribute `schmidt == True`, this is interpreted as the prefactor on gas-dependent star formation efficiency:

$$\tau_*^{-1} = \tau_{*,\text{specified}}^{-1}\left(\frac{M_\text{g}}{M_\text{g,Schmidt}}\right)^\alpha$$

where α is the power-law index on gas-dependent star formation efficiency, set by the attribute `schmidt_index`, and $\tau_{*,\text{specified}}$ is the value of this attribute.

Note: Saving functional attributes with VICE outputs requires the package dill, and extension to pickle in the Python standard library. It is recommended that VICE user's install dill >= 0.2.0.

Note: In the interstellar medium and star formation literature, this parameter is often referred to as the depletion timescale. In this documentation and in much of the galactic chemical evolution literature, it is usually referred to as the "star formation efficiency timescale."

2.2.26.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.tau_star = 1
>>> def f(t):
        if 5 <= t <= 6:
                return 1
        else:
                return 2
>>> sz.tau_star = f
```

2.2.27 vice.singlezone.dt

Type : real number

Default : 0.01

The timestep size in Gyr to use in the integration.

Note: For fine timestepping, this affects the total integration time with a dt^{-2} dependence. For coarse timestepping, the integration time is approximately constant, due to it being dominated not by timestepping but by write-out.

2.2.27.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.dt = 0.02
>>> sz.dt = 0.005
```

2.2.28 vice.singlezone.schmidt

Type : bool

Default : False

If true, the simulation will adopt a gas-dependent τ_*. At each timestep, the star formation efficiency timescale is determined via:

$$\tau_*(t) = \tau_{*,\text{specified}}(t)\left(\frac{M_g}{M_{g,\text{Schmidt}}}\right)^{-\alpha}$$

where $\tau_{*,\text{specified}}(t)$ is the value of the attribute `tau_star`, M_g is the mass of the interstellar medium, $M_{g,\text{Schmidt}}$ the normalization thereof (attribute `MgSchmidt`), and α the power-law index set by the attribute `schmidt_index`.

This is an application of the Kennicutt-Schmidt star formation law to the single-zone approximation (Kennicutt 1998[1]; Schmidt 1959[2], 1963[3]).

If False, this parameter does not impact the star formation efficiency that the user has specified.

See also:

- `vice.singlezone.tau_star`

[1] Kennicutt (1998), ApJ, 498, 541

[2] Schmidt (1959), ApJ, 129, 243

[3] Schmidt (1963), ApJ, 137, 758

- `vice.singlezone.schmidt_index`
- `vice.singlezone.MgSchmidt`

2.2.28.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.schmidt = True
>>> sz.schmidt = False
```

2.2.29 vice.singlezone.MgSchmidt

Type : real number

Default : 6.0e+09

The normalization of the gas supply in $M_\odot$ when star formation efficiency is dependent on the gas supply:

$$\tau_* \sim \left(\frac{M_g}{M_{g,\text{Schmidt}}}\right)^{-\alpha}$$

where α is specified by the attribute `schmidt_index`.

Tip: In practice, this quantity should be comparable to a typical gas supply of the simulated zone so that the actual star formation efficiency at a given timestep is near the user-specified value.

See also:

- `vice.singlezone.tau_star`
- `vice.singlezone.schmidt`
- `vice.singlezone.schmidt_index`

2.2.29.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.MgSchmidt = 5.0e+09
```

2.2.30 vice.singlezone.schmidt_index

Type : real number

Default : 0.5

The power-law index on gas-dependent star formation efficiency, if applicable:

$$\tau_*^{-1} \sim M_g^\alpha$$

Note: This number should be 1 less than the power law index which describes the scaling of star formation with the surface density of gas.

See also:

- `vice.singlezone.tau_star`
- `vice.singlezone.schmidt`
- `vice.singlezone.schmidt_index`

2.2.30.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.schmidt_index = 0.5
>>> sz.schmidt_index = 0.4
```

2.2.31 vice.singlezone.m_upper

Type : real number

Default : 100

The upper mass limit on star formation in $M_\odot$.

2.2.31.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.m_upper = 120
```

2.2.32 vice.singlezone.m_lower

Type : real number

Default : 0.08

The lower mass limit on star formation in solar masses.

2.2.32.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.m_lower = 0.1
```

2.2.33 vice.singlezone.postMS

Type : real number

Default : 0.1

New in version 1.2.0.

The ratio of a star's post main sequence lifetime to its main sequence lifetime.

2.2.33.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.postMS = 0.12
```

2.2.34 vice.singlezone.Z_solar

Type : real number

Default : 0.014

The metallicity by mass of the sun $M_Z/M_\odot$. This is used in calibrating the total metallicity of the ISM, which is necessary when there are only a few elements tracked by the simulation with metallicity dependent yields. This scaling is implemented as follows:

$$Z_{\text{ISM}} = Z_\odot \left[\sum_i Z_i\right] \left[\sum_i Z_i^\odot\right]^{-1}$$

where the summation is taken over the elements tracked by the simulation.

Note: The default value is the metallicity calculated by Asplund et al. (2009)[1]. VICE adopts the Asplund et al. (2009) measurements on their element-by-element basis in calculating [X/H] and [X/Y] in simulations; it is thus recommended that users adopt these measurements as well so that the adopted solar composition is self-consistent. This however has no qualitative impact on the behavior of the simulation.

2.2.34.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example")
>>> sz.Z_solar = 0.014
```

2.2.35 vice.singlezone.agb_model

[DEPRECATED]

Type : `str` [case-insensitive]

Default : None

[1] Asplund et al. (2009), ARA&A, 47, 481

Deprecated since version 1.2.0: Users should instead use the `vice.yields.agb.settings dataframe` to declare their yields. These allow the same keywords as this attribute as well as user-constructed functions of stellar mass and metallicity.

A keyword denoting which stellar mass-metallicity grid of fractional nucleosynthetic yields from asymptotic giant branch (AGB) stars to adopt.

Recognized Keywords:

- "cristallo11"[1]
- "karakas10"[2]

Note: If the Karakas (2010) set of yields are adopted and any elements tracked by the simulation are heavier than nickel, a LookupError will be raised. The Karakas (2010) study did not report yields for elements heavier than nickel.

2.2.35.1 Example Code

```
>>> import vice
>>> sz = vice.singlezone(name = "example", elements = ["c", "n", "o"])
>>> sz.agb_model = "karakas10"
>>> sz.agb_model = "cristallo11"
```

[1] Cristallo et al. (2011), ApJS, 197, 17

[2] Karakas (2010), MNRAS, 403, 1413