DUSTGROWTH IN PHANTOM

Arnaud Vericel, Jean-François Gonzalez

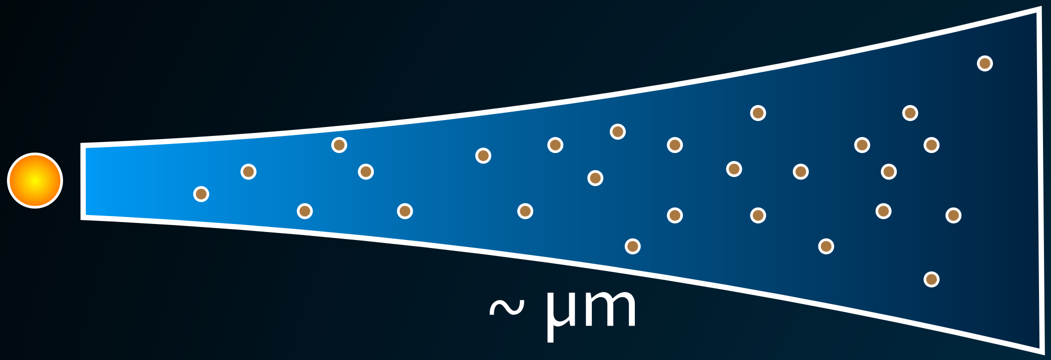
CRAL
CENTRE DE RECHERCHE ASTROPHYSIQUE DE LYON

UCLB
observatoire de Lyon

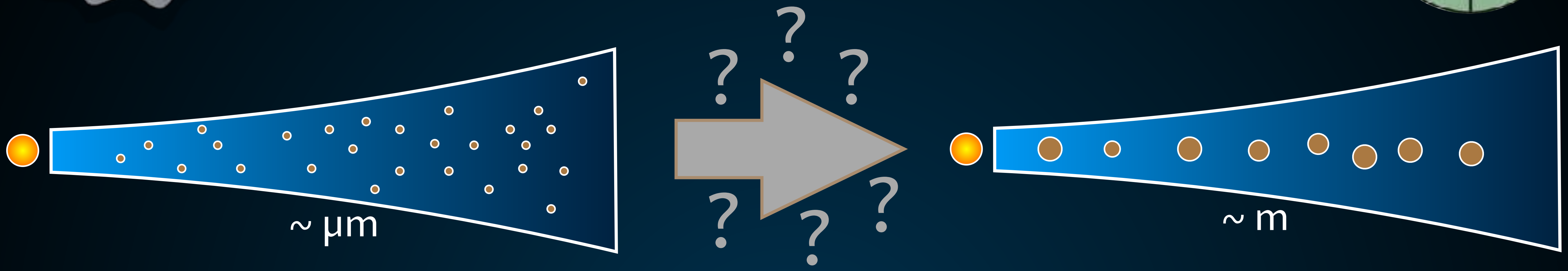
PHANTOM

CONTEXT

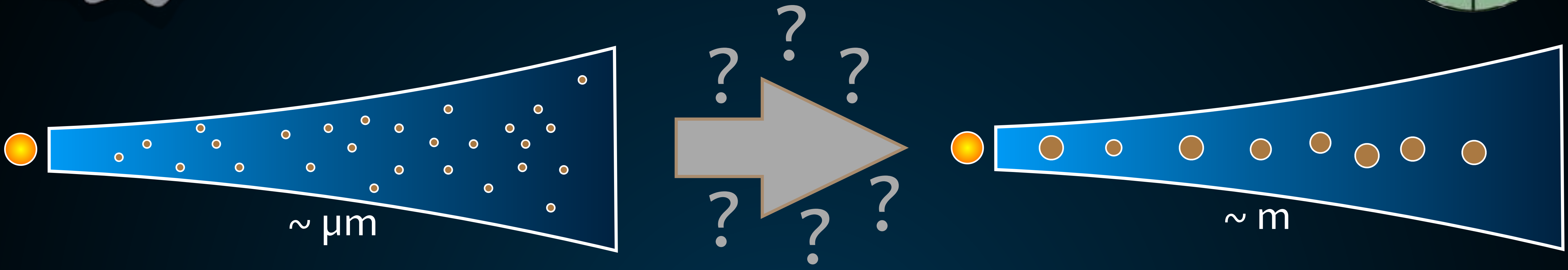
10 %



CONTEXT



CONTEXT



MODEL

- Smoluchowski equation (1916) (Maxime Lombart’s PhD project)

$$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^m K(m', m - m')n(m')n(m - m')\mathrm{d}m' - n(m)\int_0^{+\infty} K(m', m)n(m')\mathrm{d}m'$$

MODEL

- ★ Smoluchowski equation (1916) (Maxime Lombart's PhD project)

$$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^m$$

challenging and expensive

$$m', m)n(m')\mathrm{d}m'$$

MODEL

20 %

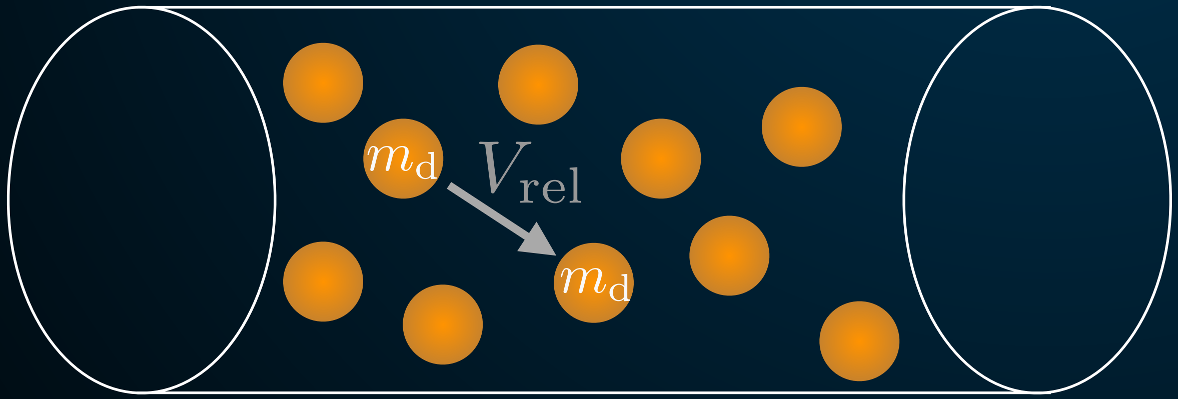
- Smoluchowski equation (1916) (Maxime Lombart's PhD project)

$$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^{m}$$

challenging and expensive

$$m', m)n(m')\mathrm{d}m'$$

- Stepinski & Valageas (1997) : pure growth



MODEL

- Smoluchowski equation (1916) (Maxime Lombart's PhD project)

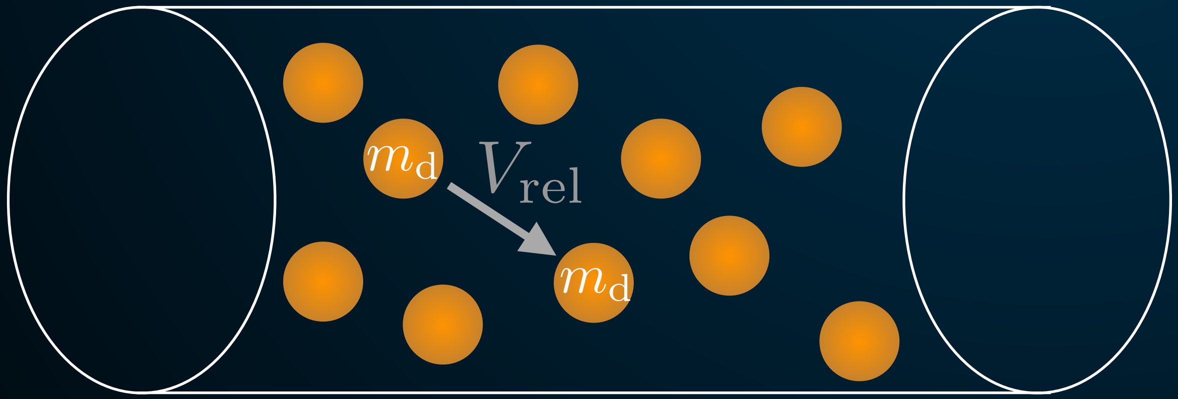
$$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^{m}$$

challenging and expensive

$$m', m)n(m')\mathrm{d}m'$$

- Stepinski & Valageas (1997) : pure growth

$$\frac{\mathrm{d}m_{\mathrm{d}}}{\mathrm{d}t} = \frac{m_{\mathrm{d}}}{t_{\mathrm{col}}}$$



MODEL

PHANTOM

20 %

- Smoluchowski equation (1916) (Maxime Lombart's PhD project)

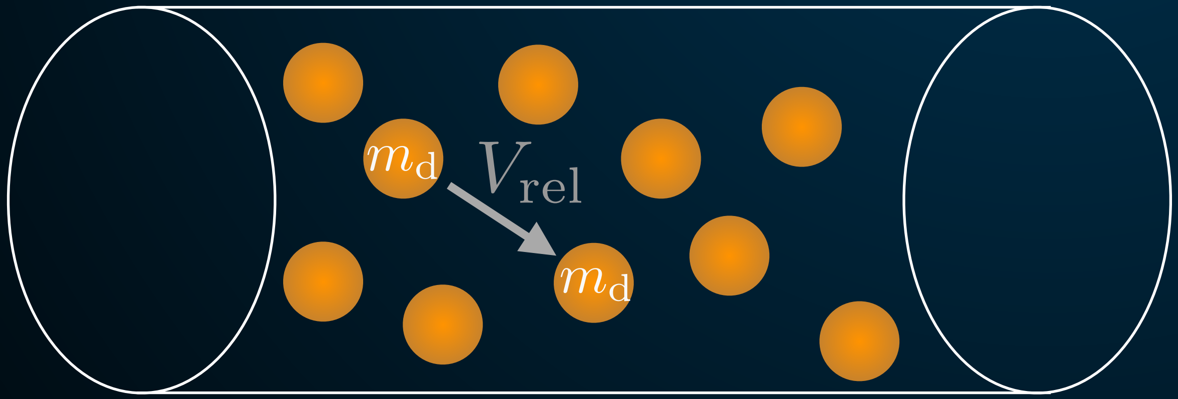
$$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^{m}$$

challenging and expensive

$$m', m)n(m')\mathrm{d}m'$$

- Stepinski & Valageas (1997) : pure growth

$$\frac{\mathrm{d}m_{\mathrm{d}}}{\mathrm{d}t} = \frac{m_{\mathrm{d}}}{t_{\mathrm{col}}}$$



$$t_{\mathrm{col}} = \frac{1}{\sigma V_{\mathrm{rel}} n_{\mathrm{d}}}$$

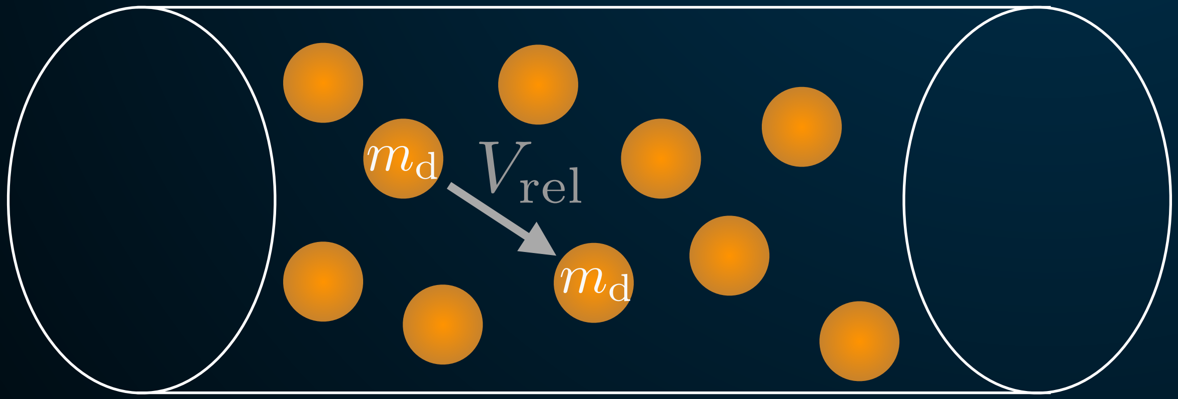
MODEL

- Smoluchowski equation (1916) (Maxime Lombart's PhD project)

$$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^{m}$$ ***challenging and expensive*** $$m', m)n(m')\mathrm{d}m'$$

- Stepinski & Valageas (1997) : pure growth

$$\frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\rho_\mathrm{d}}{\rho_\mathrm{s}} V_\mathrm{rel}$$



$$t_\mathrm{col} = \frac{1}{\sigma V_\mathrm{rel} n_\mathrm{d}}$$

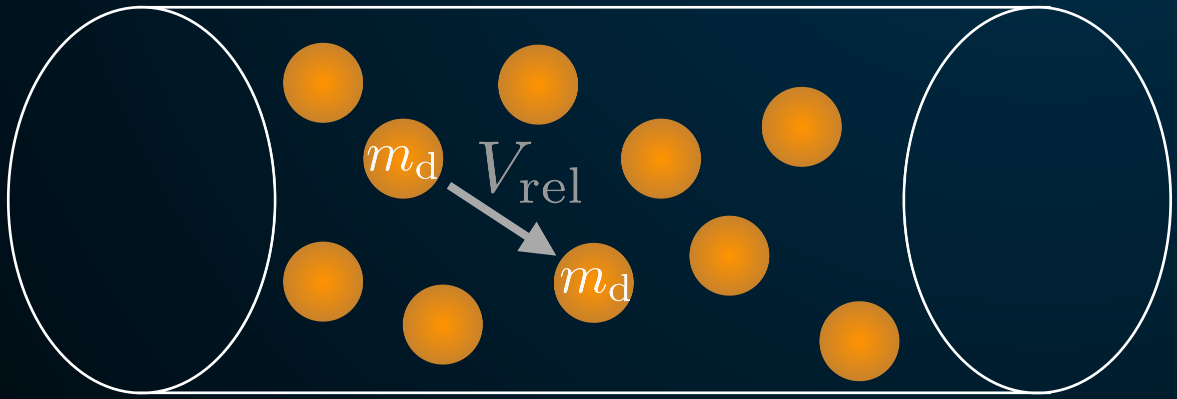
MODEL

- Smoluchowski equation (1916) (Maxime Lombart's PhD project)

$\frac{\partial n(m)}{\partial t} = \frac{1}{2}\int_0^{m}$ **challenging and expensive** $m', m)n(m')\mathrm{d}m'$

- Stepinski & Valageas (1997) : pure growth

$$\frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\rho_{\mathrm{d}}}{\rho_{\mathrm{s}}} V_{\mathrm{rel}}$$



$$t_{\mathrm{col}} = \frac{1}{\sigma V_{\mathrm{rel}} n_{\mathrm{d}}}$$

$$\boxed{\begin{gathered} V_{\mathrm{rel}} = \sqrt{2^{3/2}\mathrm{Ro}\alpha c_{\mathrm{s}}}\frac{\sqrt{\mathrm{Sc}-1}}{\mathrm{Sc}} \\ \mathrm{Sc} = 1 + \Omega_{\mathrm{K}} t_{\mathrm{s}} \sqrt{1 + \frac{\Delta v^2}{V_{\mathrm{t}}^2}} \end{gathered}}$$

MODEL

MODEL

- Fragmentation

We compare V_{rel} with a fragmentation velocity V_{frag}

MODEL

- Fragmentation

We compare $V_{\rm rel}$ with a fragmentation velocity $V_{\rm frag}$

$$V_{\rm rel} < V_{\rm frag} : \frac{{\rm d}s}{{\rm d}t} = \frac{\rho_{\rm d}}{\rho_{\rm s}} V_{\rm rel} \quad \text{: growth}$$

MODEL

- Fragmentation

We compare V_{rel} with a fragmentation velocity V_{frag}

$$V_{\mathrm{rel}} < V_{\mathrm{frag}} : \frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\rho_{\mathrm{d}}}{\rho_{\mathrm{s}}} V_{\mathrm{rel}} \quad \text{: growth}$$

$$V_{\mathrm{rel}} > V_{\mathrm{frag}} : \frac{\mathrm{d}s}{\mathrm{d}t} = -\frac{\rho_{\mathrm{d}}}{\rho_{\mathrm{s}}} V_{\mathrm{rel}} f \quad \text{: fragmentation}$$

MODEL

★ Fragmentation

We compare $V_{\rm rel}$ with a fragmentation velocity $V_{\rm frag}$

$$V_{\rm rel} < V_{\rm frag} \; : \; \frac{{\rm d}s}{{\rm d}t} = \frac{\rho_{\rm d}}{\rho_{\rm s}} V_{\rm rel} \quad \text{: growth}$$

$$V_{\rm rel} > V_{\rm frag} \; : \; \frac{{\rm d}s}{{\rm d}t} = -\frac{\rho_{\rm d}}{\rho_{\rm s}} V_{\rm rel} f \quad \text{: fragmentation}$$

$f = 1$: catastrophic disruption

$f = \dfrac{V_{\rm rel}^2}{V_{\rm rel}^2 + V_{\rm frag}^2}$: smoother fragmentation derived from Kobayashi & Tanaka (2009)

MODEL

MODEL

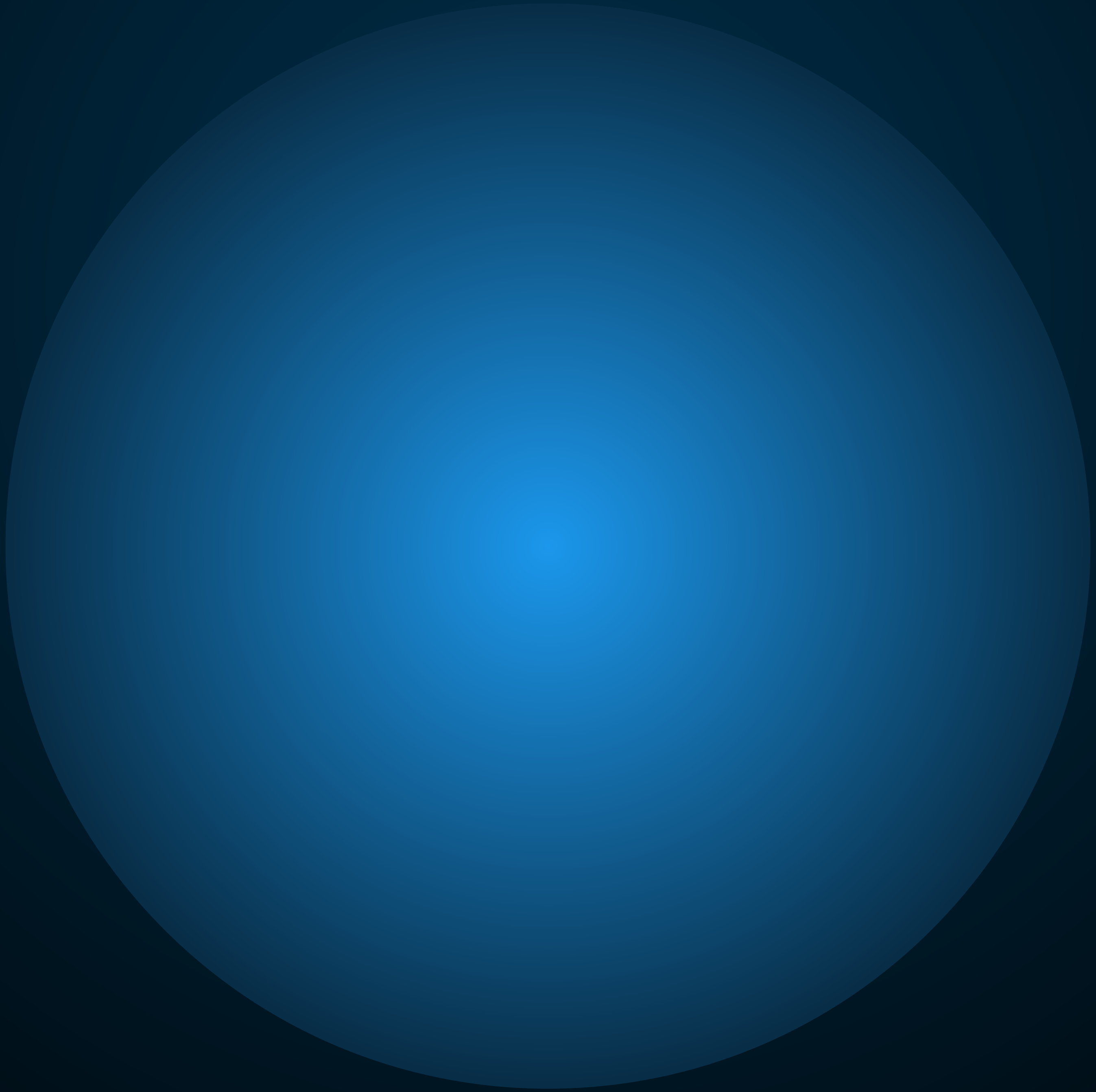
Snow lines

We model snow lines by changing V_{frag} between 2 parts of the disk

MODEL

- Snow lines

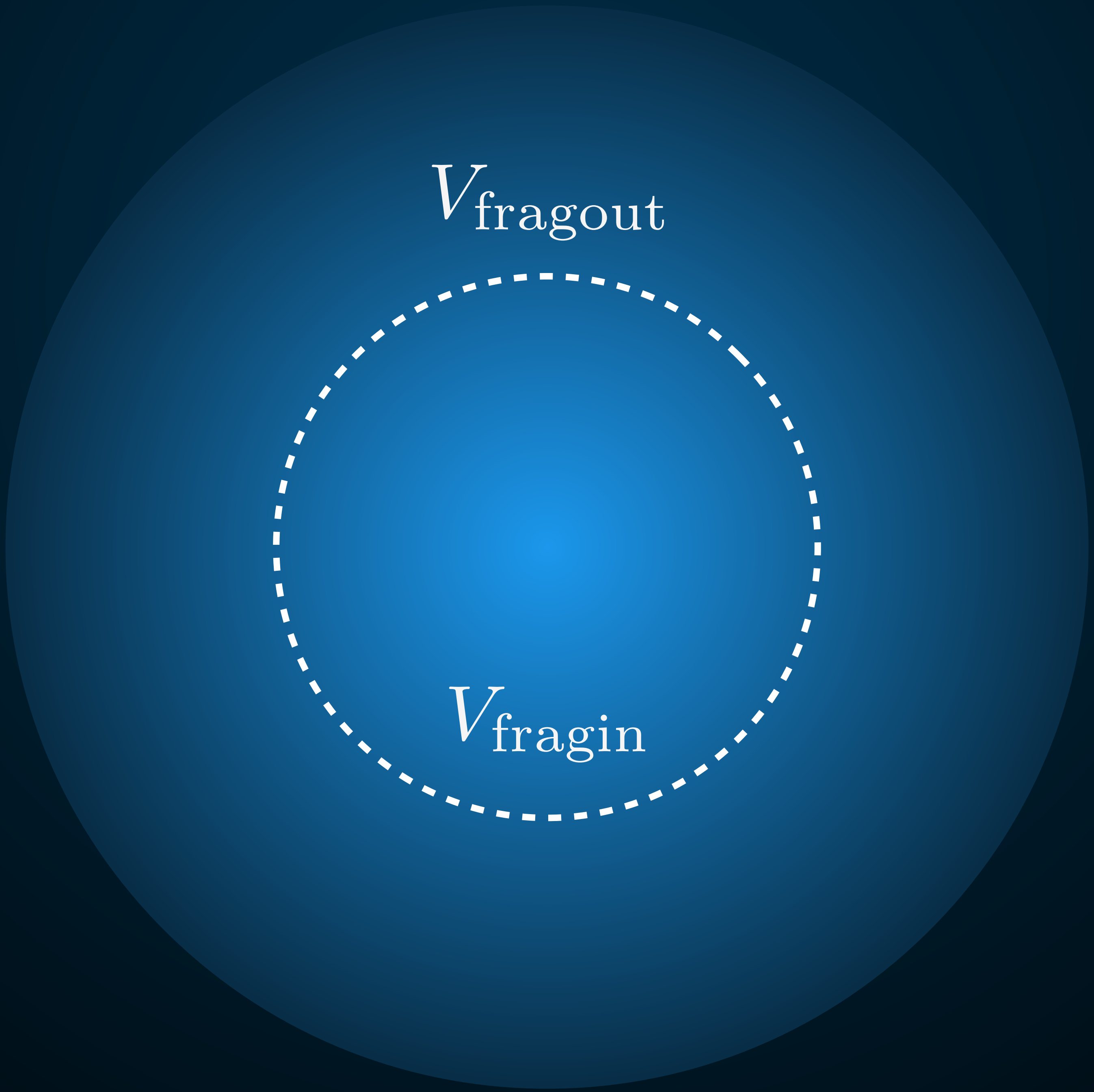
We model snow lines by changing V_{frag} between 2 parts of the disk



MODEL

★ Snow lines

We model snow lines by changing V$_{frag}$ between 2 parts of the disk

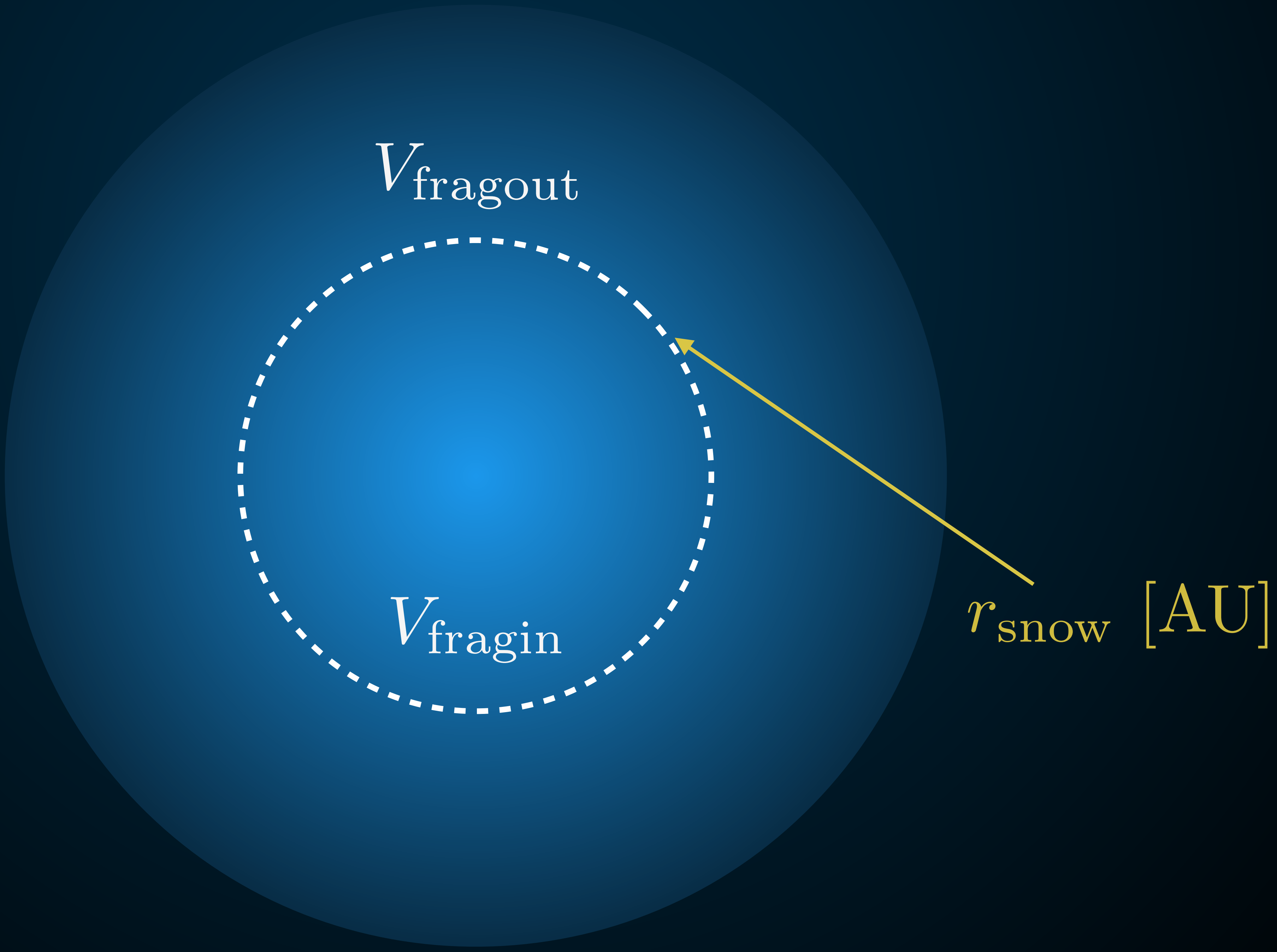


MODEL

40 %

★ Snow lines

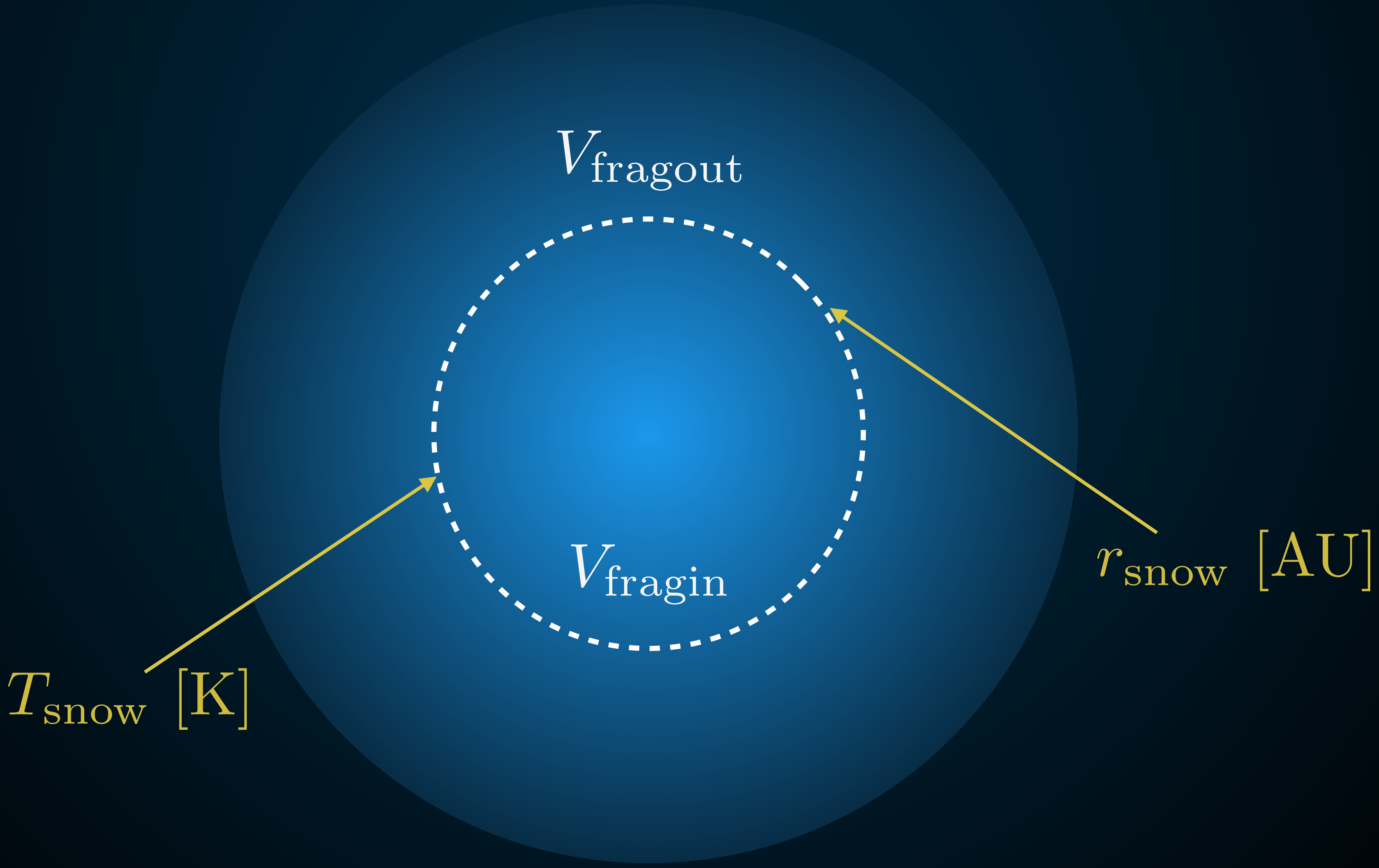
We model snow lines by changing V_{frag} between 2 parts of the disk



MODEL

★ Snow lines

We model snow lines by changing V_{frag} between 2 parts of the disk



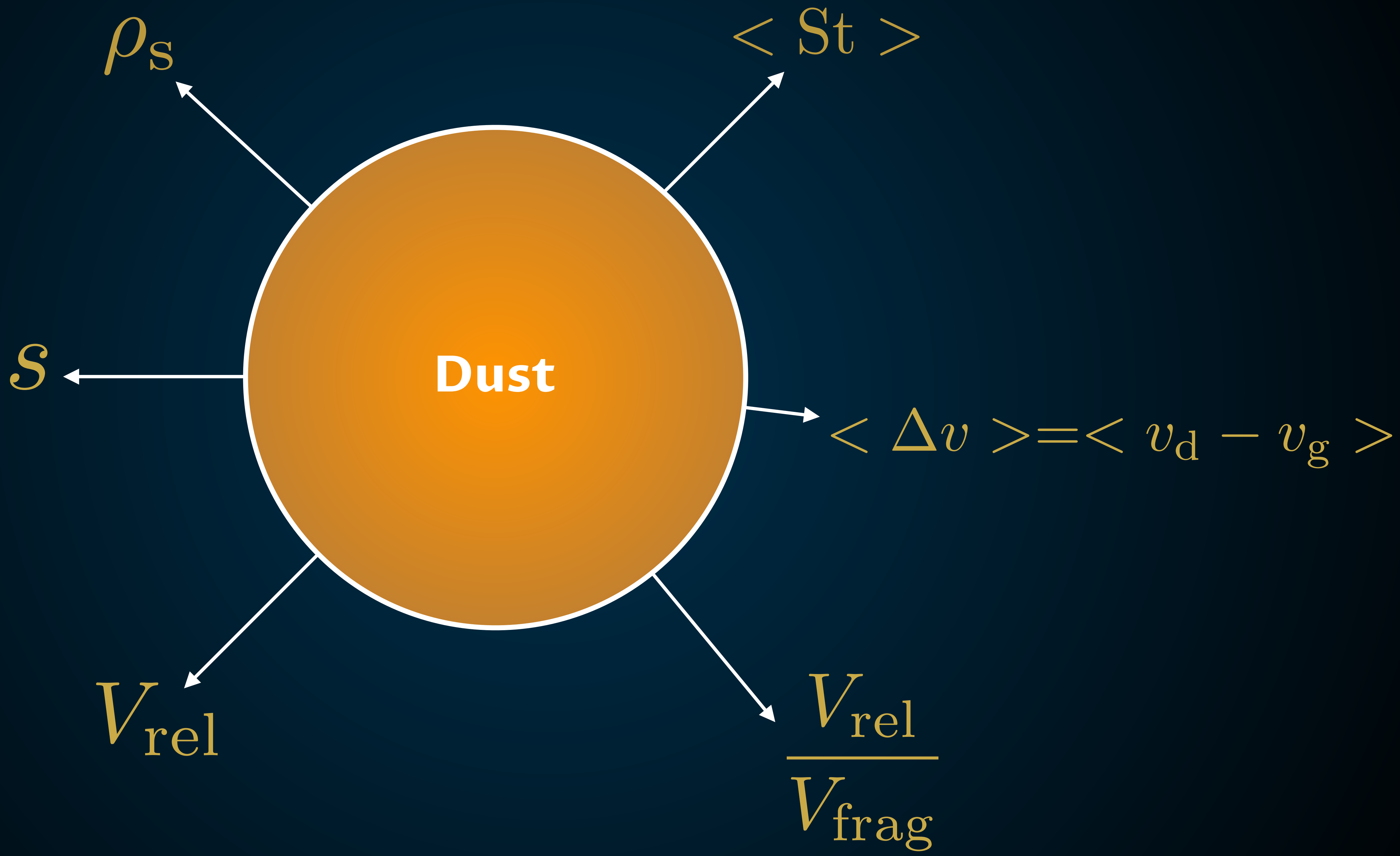
IMPLEMENTATION

50 %

Dust

IMPLEMENTATION

50 %



Dust

$\rho_{\rm s}$

$< {\rm St} >$

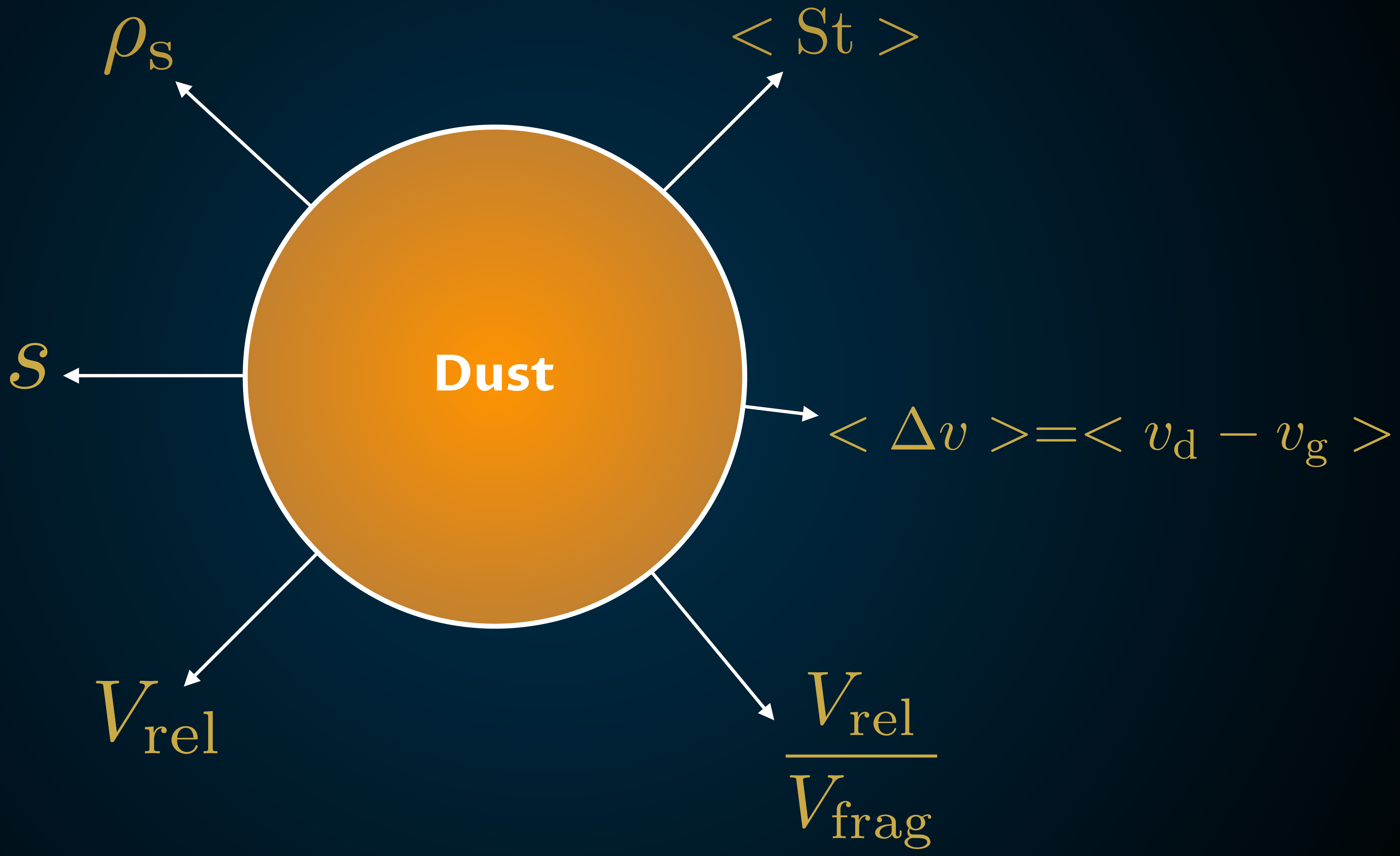
s

$< \Delta v > = < v_{\rm d} - v_{\rm g} >$

$V_{\rm rel}$

$\dfrac{V_{\rm rel}}{V_{\rm frag}}$

IMPLEMENTATION



Dust

ρ_{s}

$< \mathrm{St} >$

s

$< \Delta v > = < v_{\mathrm{d}} - v_{\mathrm{g}} >$

V_{rel}

$\dfrac{V_{\mathrm{rel}}}{V_{\mathrm{frag}}}$

New array ‘dustprop’ and new module ‘growth’

IMPLEMENTATION

60 %

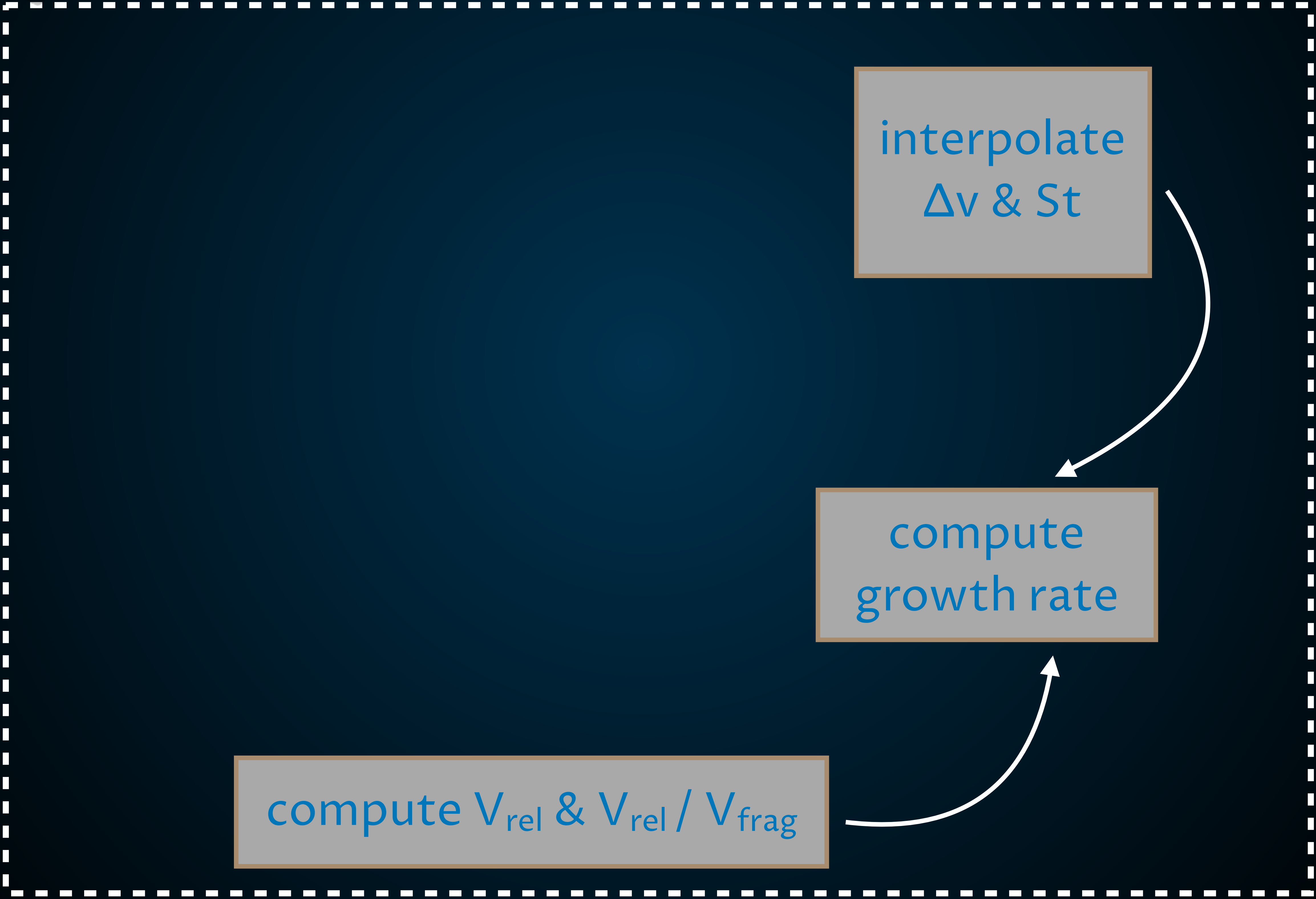
IMPLEMENTATION

60 %

interpolate
Δv & St

IMPLEMENTATION

60 %



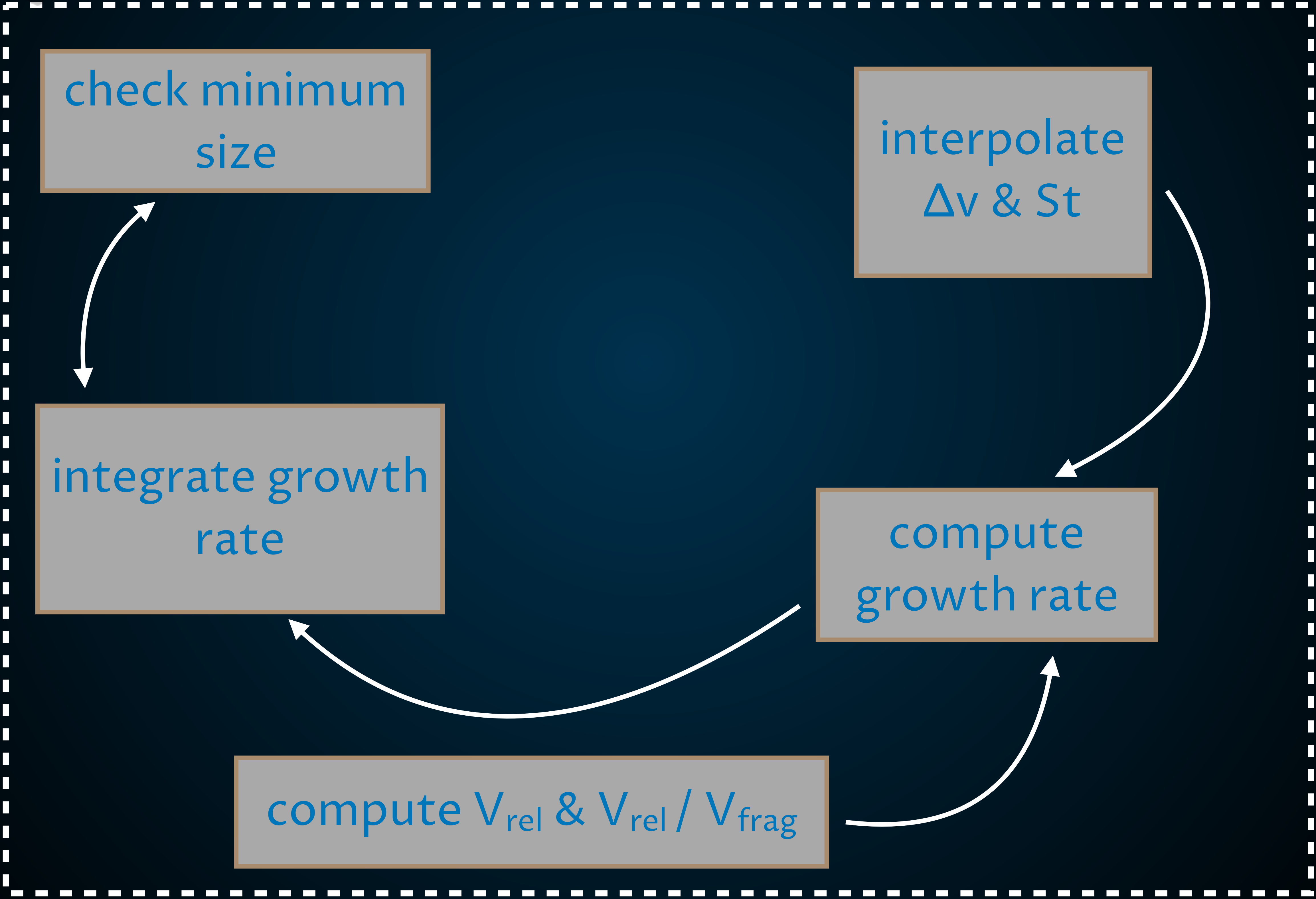
interpolate Δv & St

compute growth rate

compute V_{rel} & V_{rel} / V_{frag}

IMPLEMENTATION

60 %



check minimum size

interpolate Δv & St

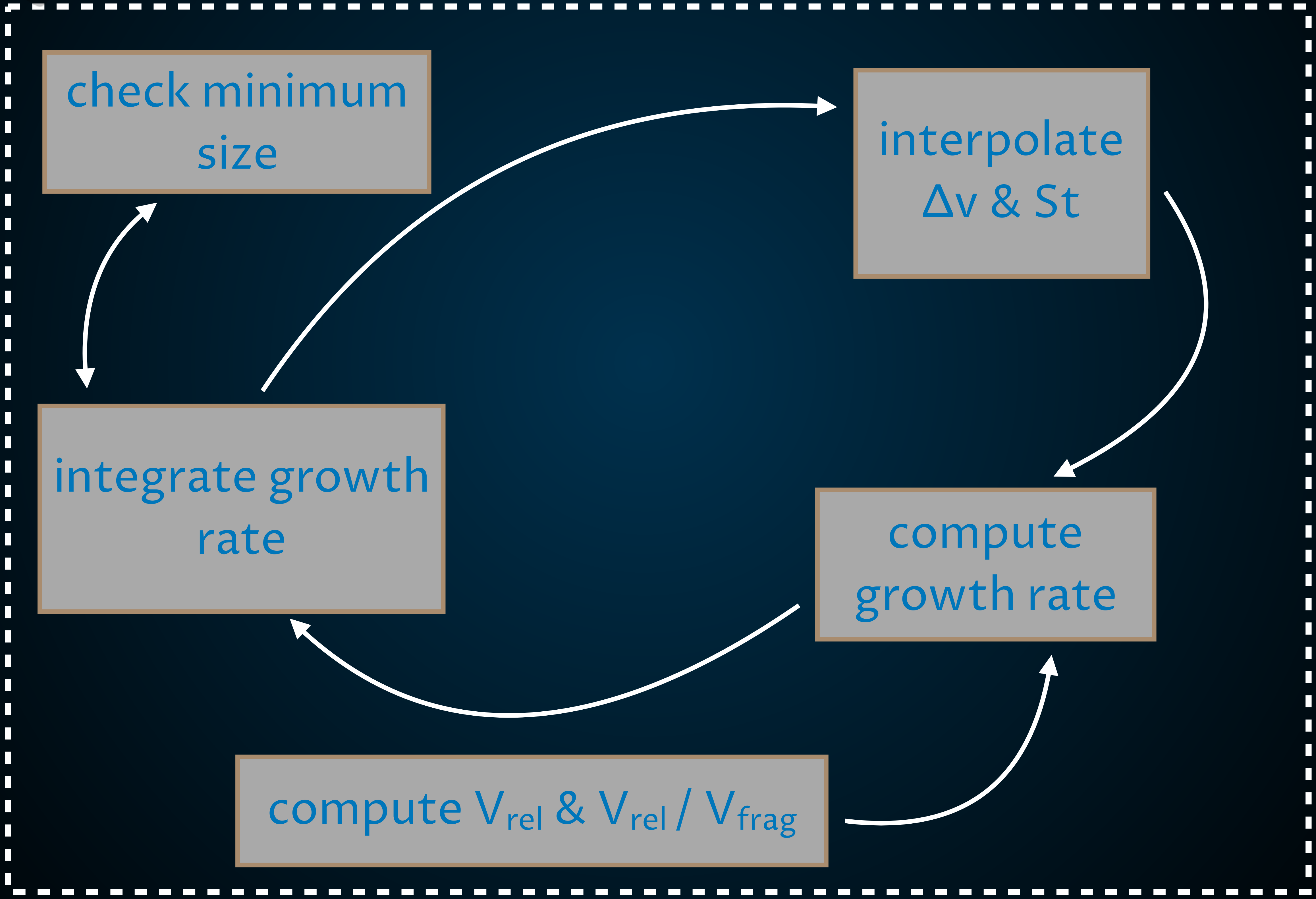
integrate growth rate

compute growth rate

compute V_{rel} & V_{rel} / V_{frag}

IMPLEMENTATION

60 %



check minimum size

interpolate Δv & St

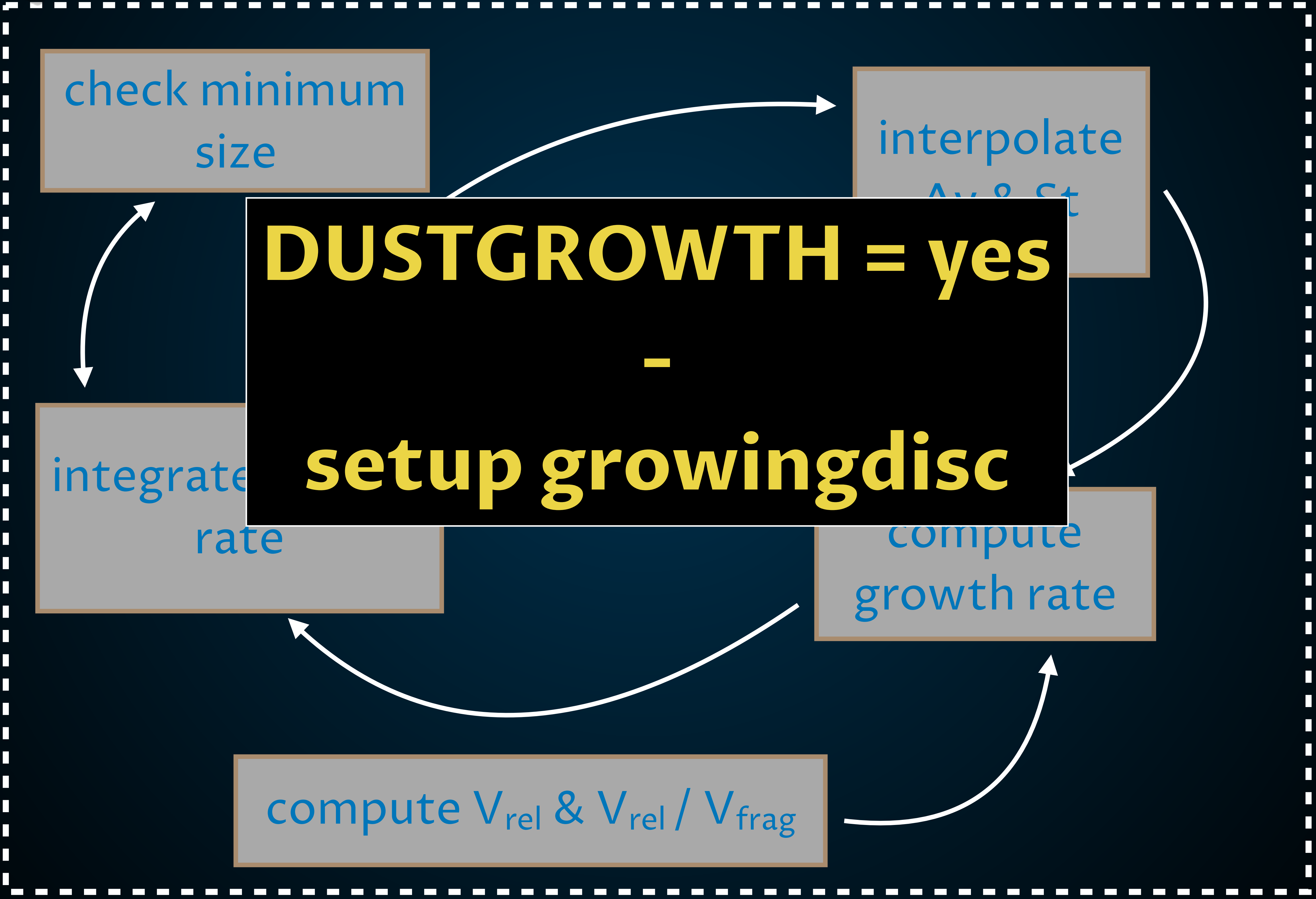
compute growth rate

integrate growth rate

compute V_{rel} & V_{rel} / V_{frag}

IMPLEMENTATION

60 %



check minimum size

interpolate Δv & St

compute growth rate

integrate rate

compute V_{rel} & V_{rel} / V_{frag}

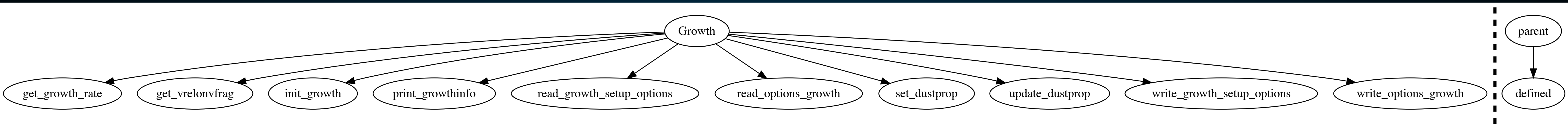
DUSTGROWTH = yes

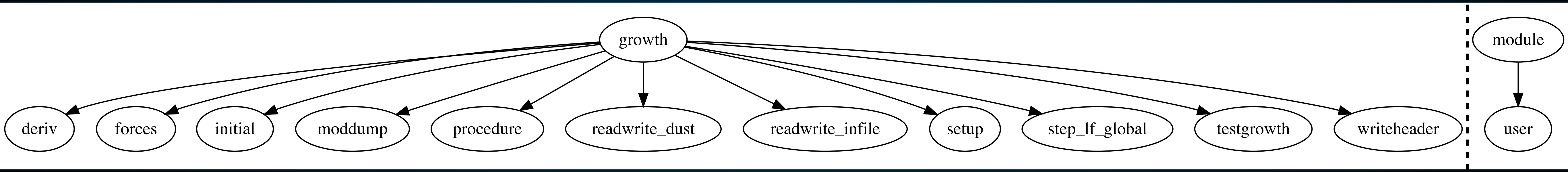
-

setup growingdisc

FORTREE MAP

70 %





GROWINGBOX

80 %

The features included in the growth module are tested

GROWINGBOX

The features included in the growth module are tested

—> check the size integration for several configurations

GROWINGBOX

80 %

The features included in the growth module are tested

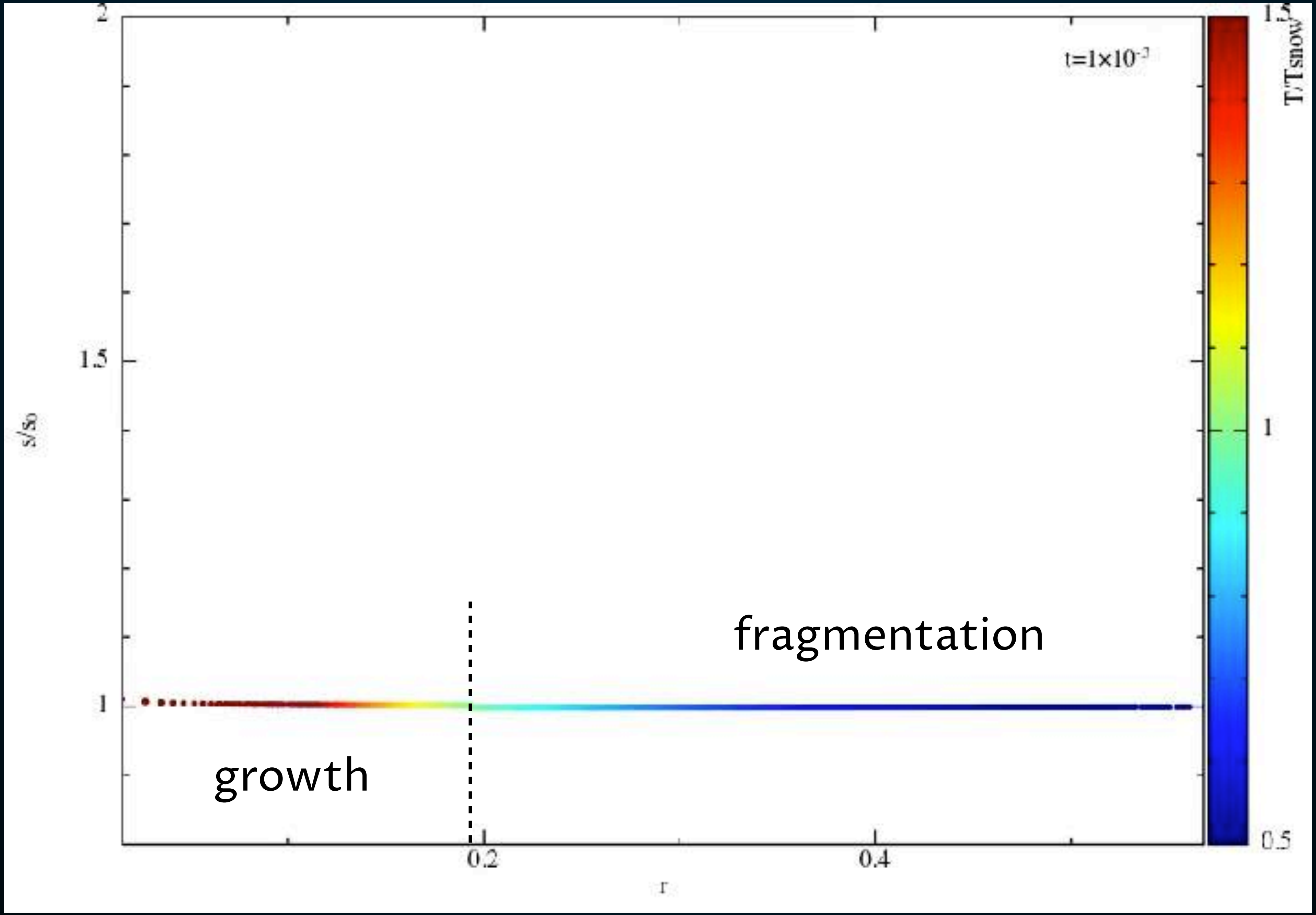
—> check the size integration for several configurations
—> check the interpolation of St

GROWINGBOX

80 %

The features included in the growth module are tested

—> check the size integration for several configurations
—> check the interpolation of St

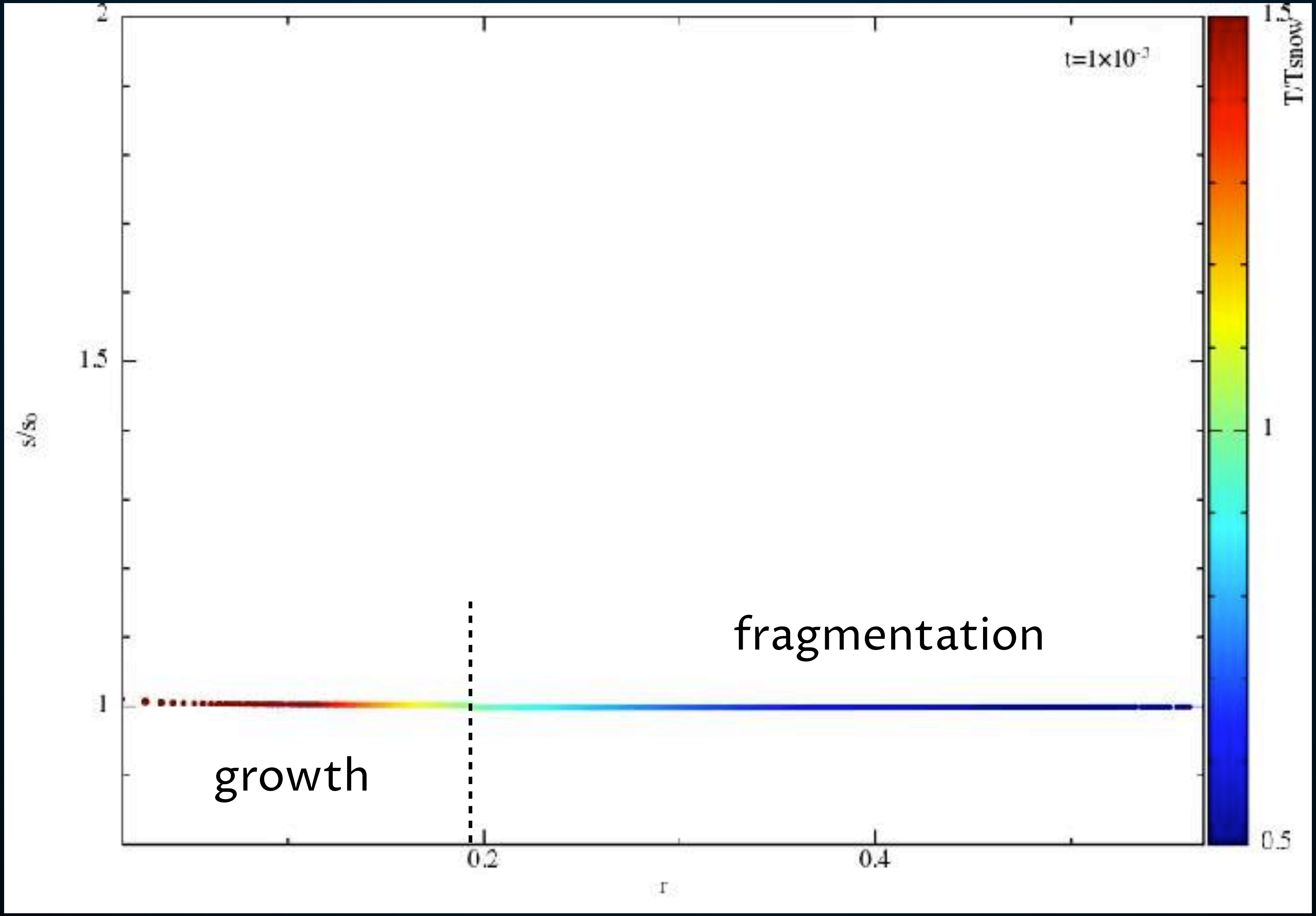


GROWINGBOX

80 %

The features included in the growth module are tested

—> check the size integration for several configurations
—> check the interpolation of St

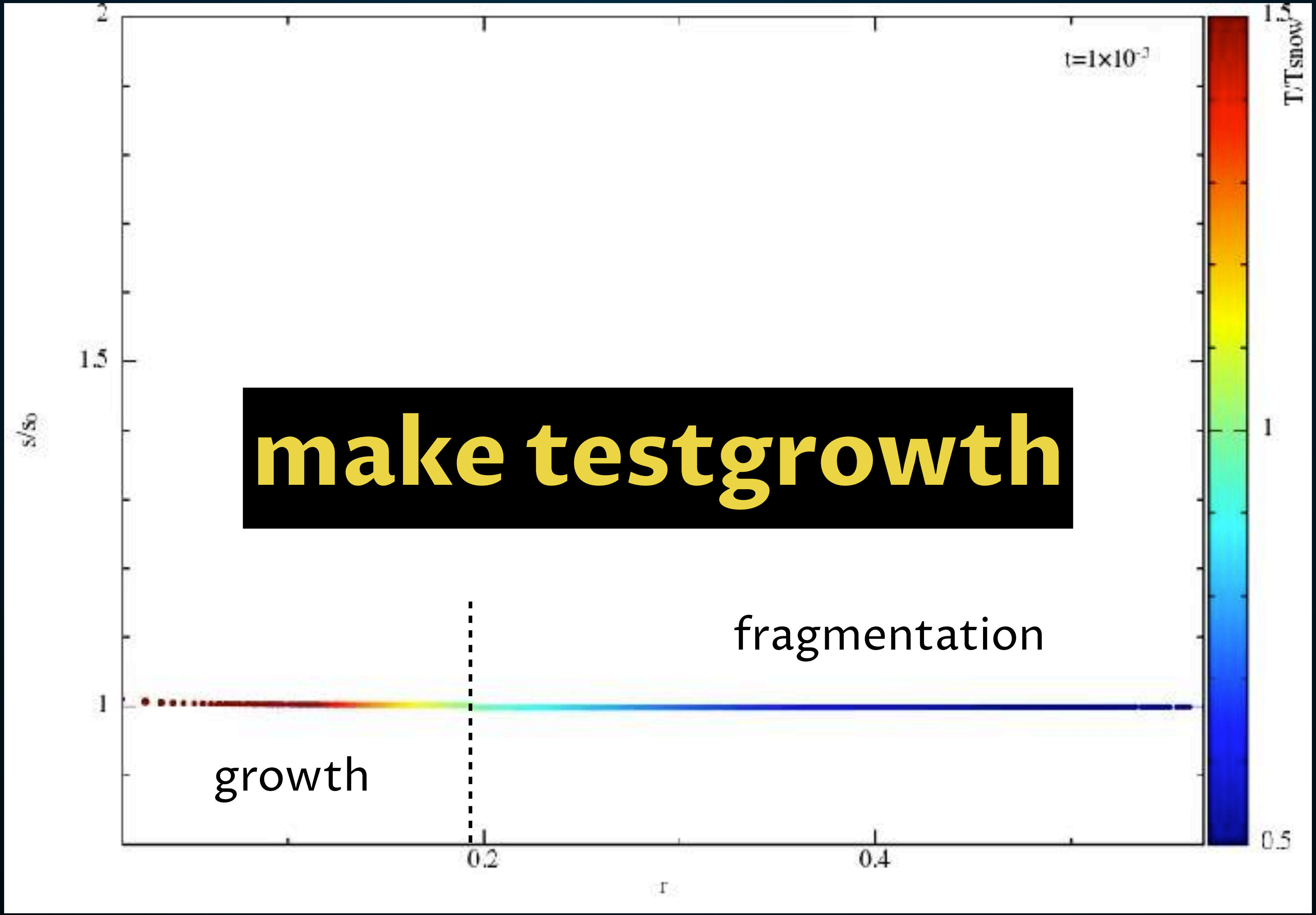


GROWINGBOX

80 %

The features included in the growth module are tested

—> check the size integration for several configurations
—> check the interpolation of St



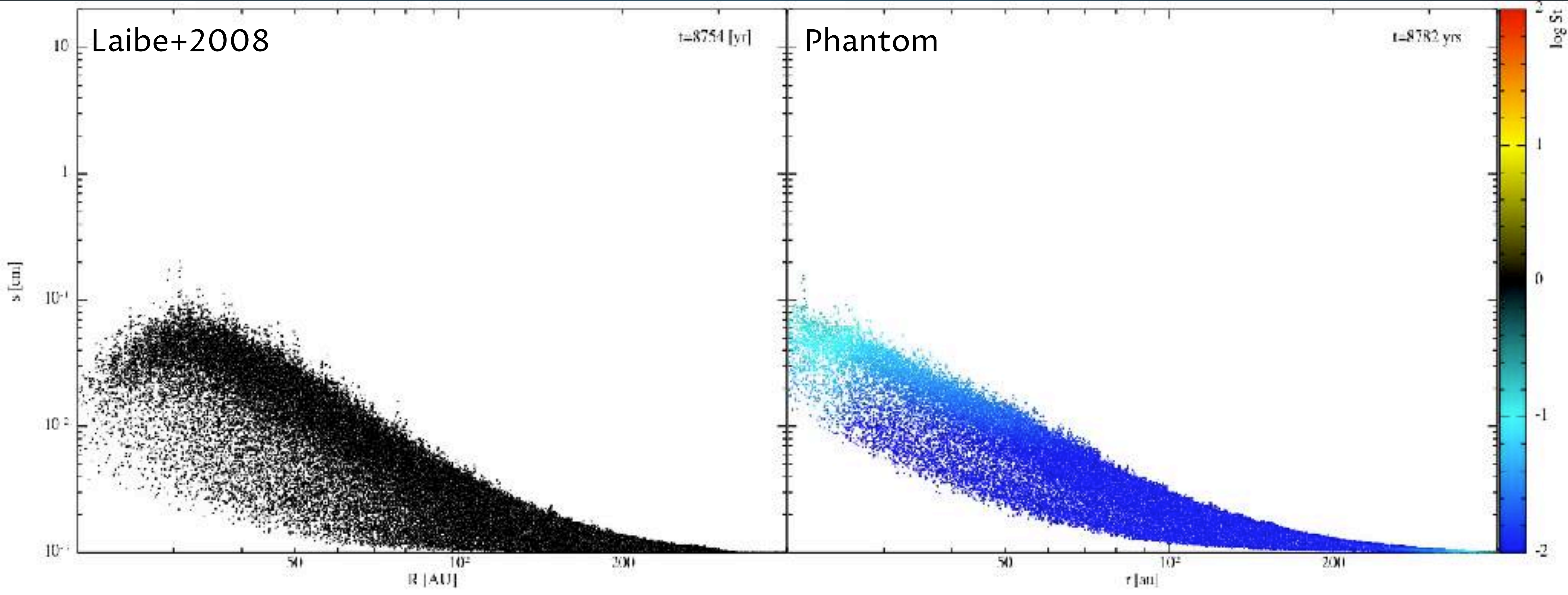
make testgrowth

COMPARISON RESULTS

90 %

p	q (c_s)	R_{in}	R_{out}	m_{disc}/m_{sun}	ε_0	H/R @ 100 AU	s_0
3/2	3/8	20 AU	300 AU	0.01	0.01	0.05	10 μm

Laibe et al. 2008 (pure growth)

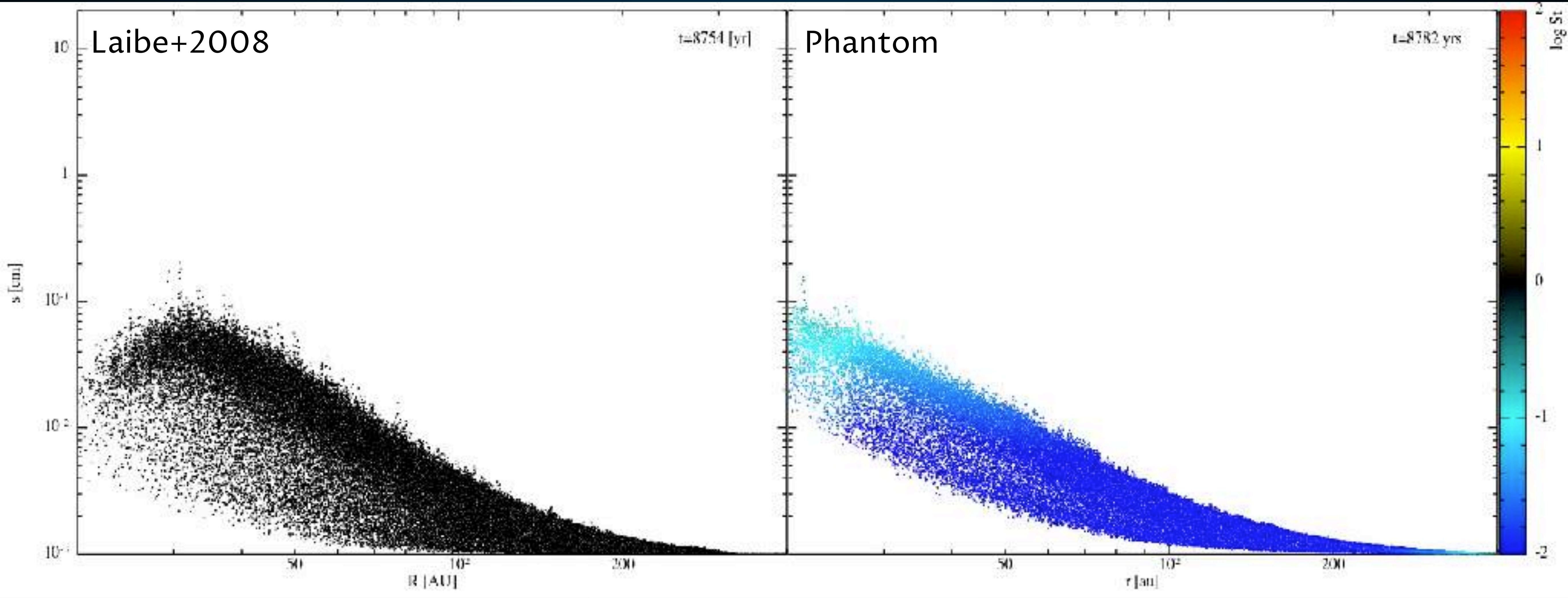


COMPARISON RESULTS

90 %

p	q (c_s)	R_{in}	R_{out}	m_{disc}/m_{sun}	ε_0	H/R @ 100 AU	s_0
3/2	3/8	20 AU	300 AU	0.01	0.01	0.05	10 µm

Laibe et al. 2008 (pure growth)

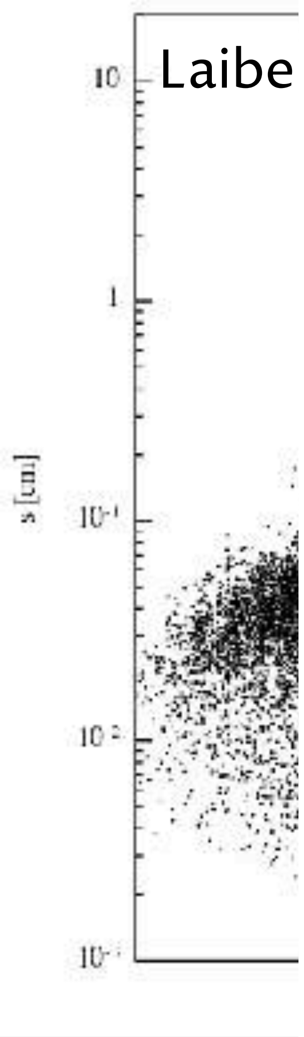


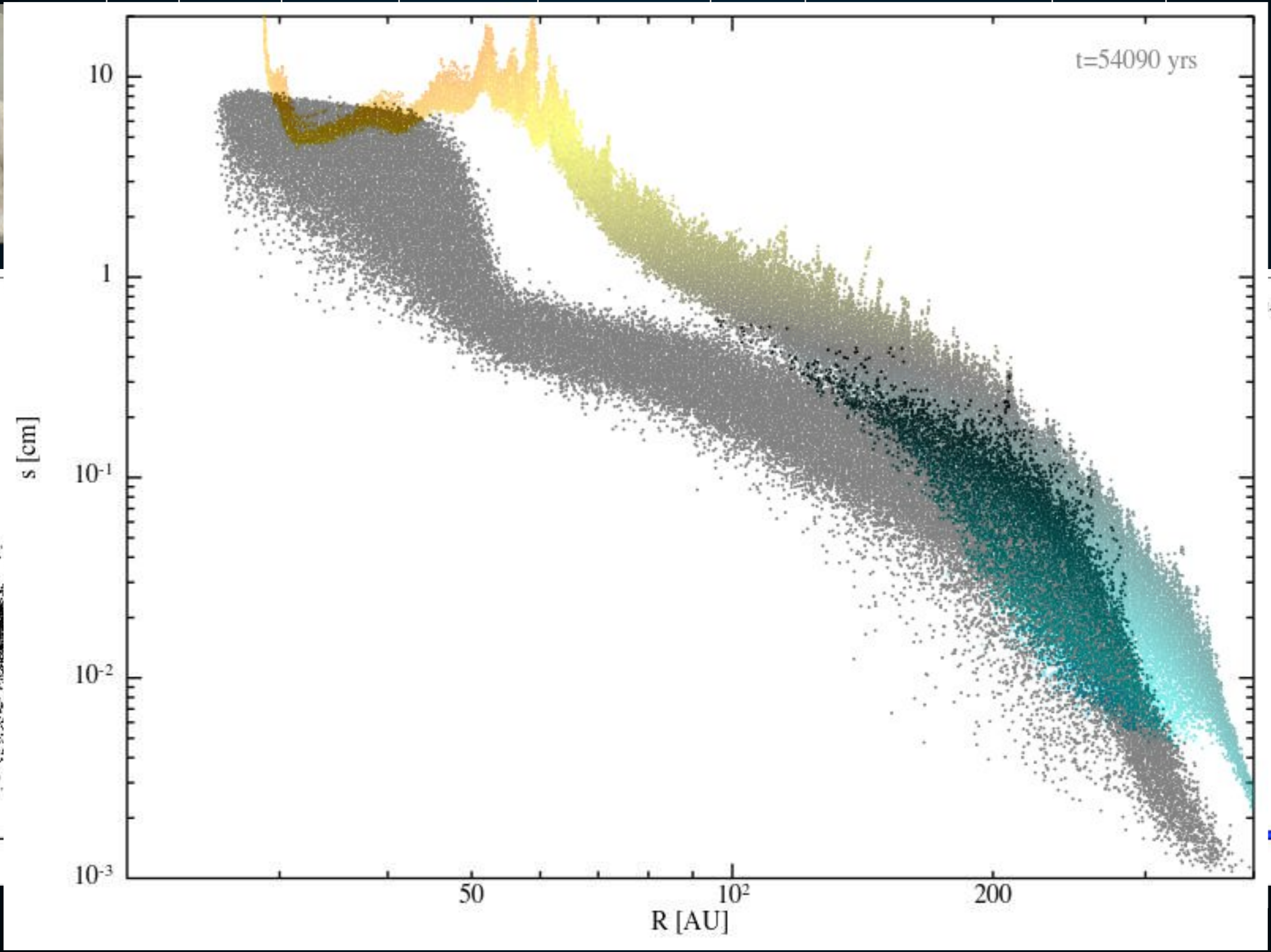
COMPARISON RESULTS

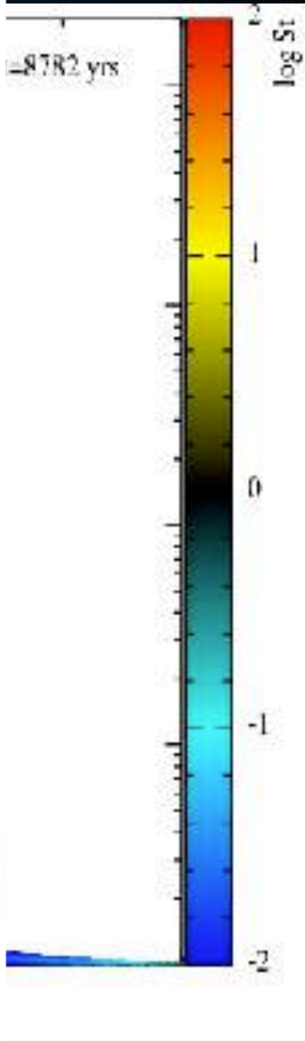
90 %

p	q (c_s)	R_{in}	R_{out}	m_{disc}/m_{sun}	ε_0	H/R @ 100 AU	s_0

Laibe



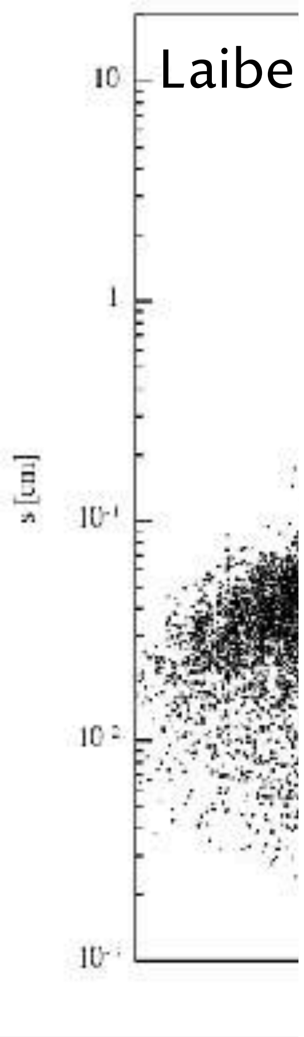


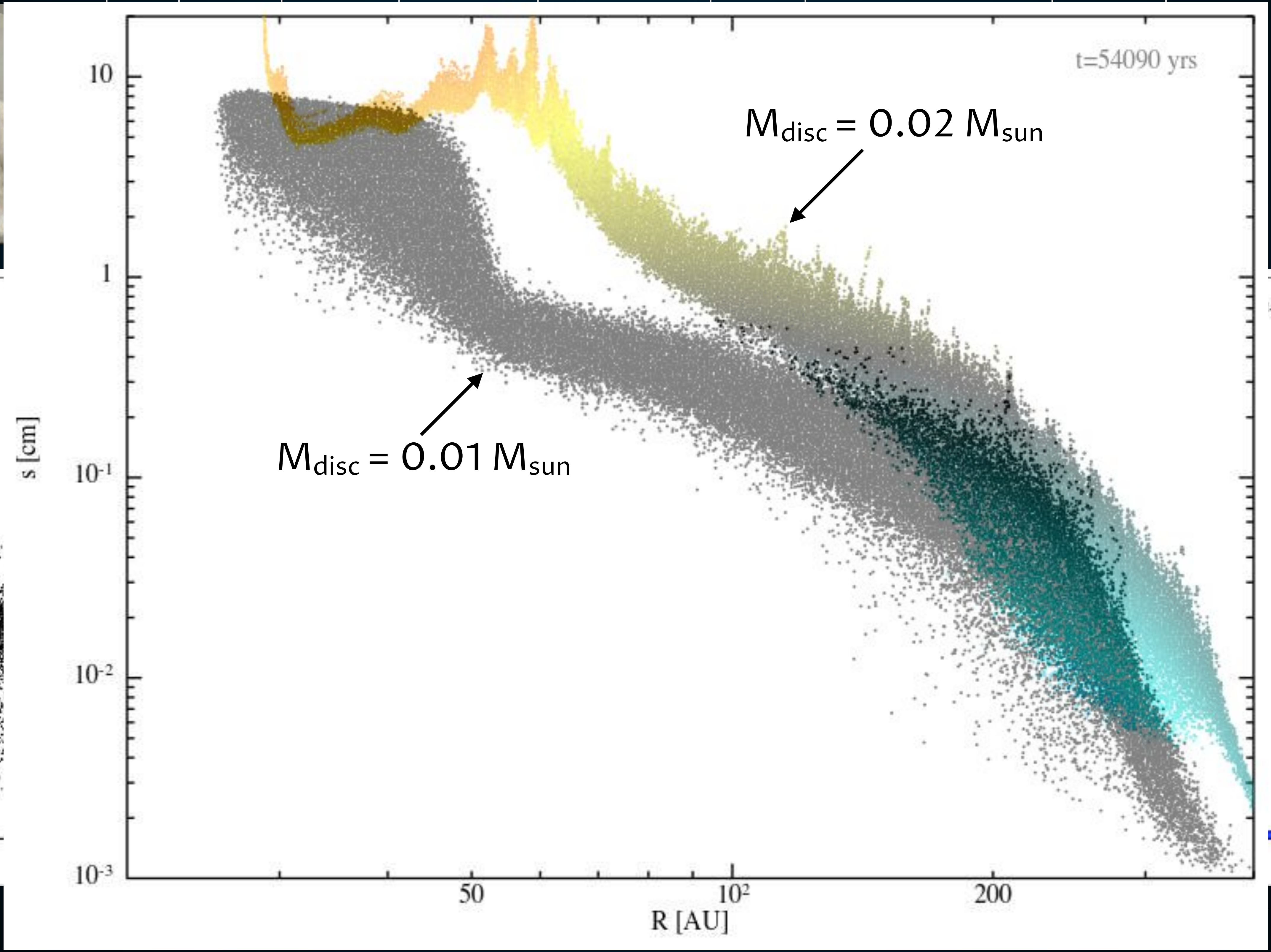


COMPARISON RESULTS

90 %

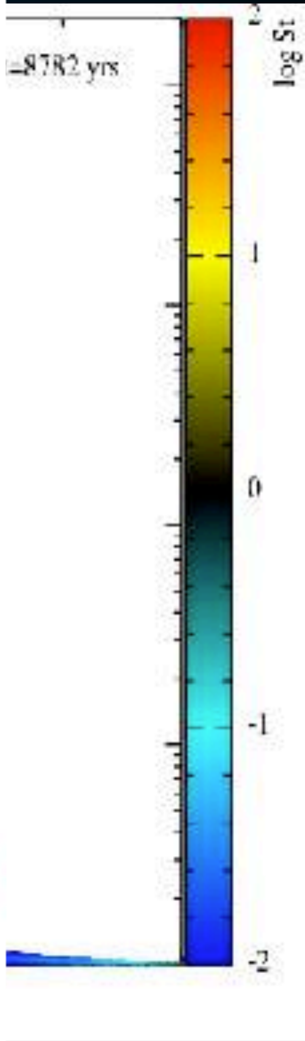
p	q (c_s)	R_{in}	R_{out}	m_{disc}/m_{sun}	ε_0	H/R @ 100 AU	s_0





$M_{disc} = 0.02\ M_{sun}$

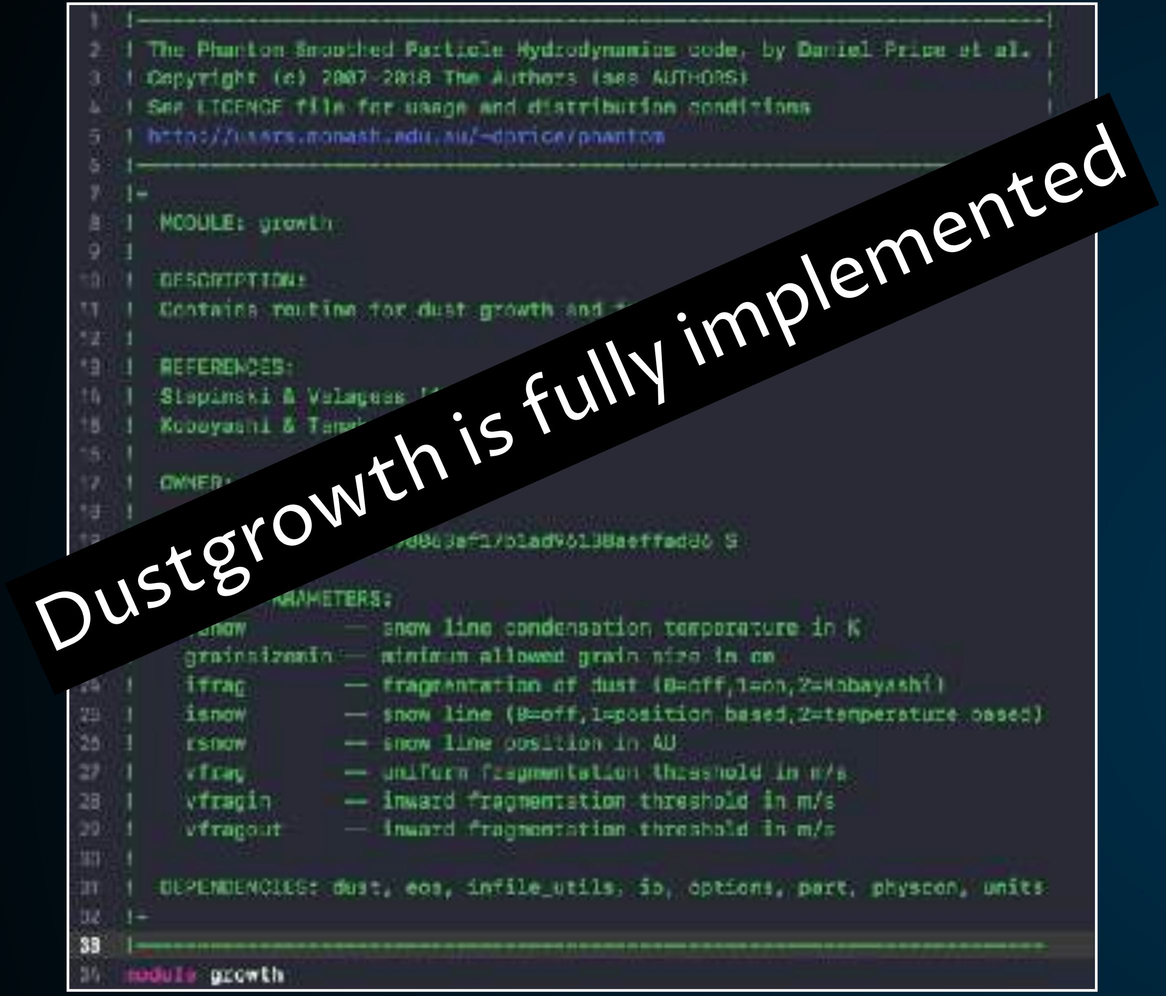
$M_{disc} = 0.01\ M_{sun}$



CONCLUSION

CONCLUSION

100 %

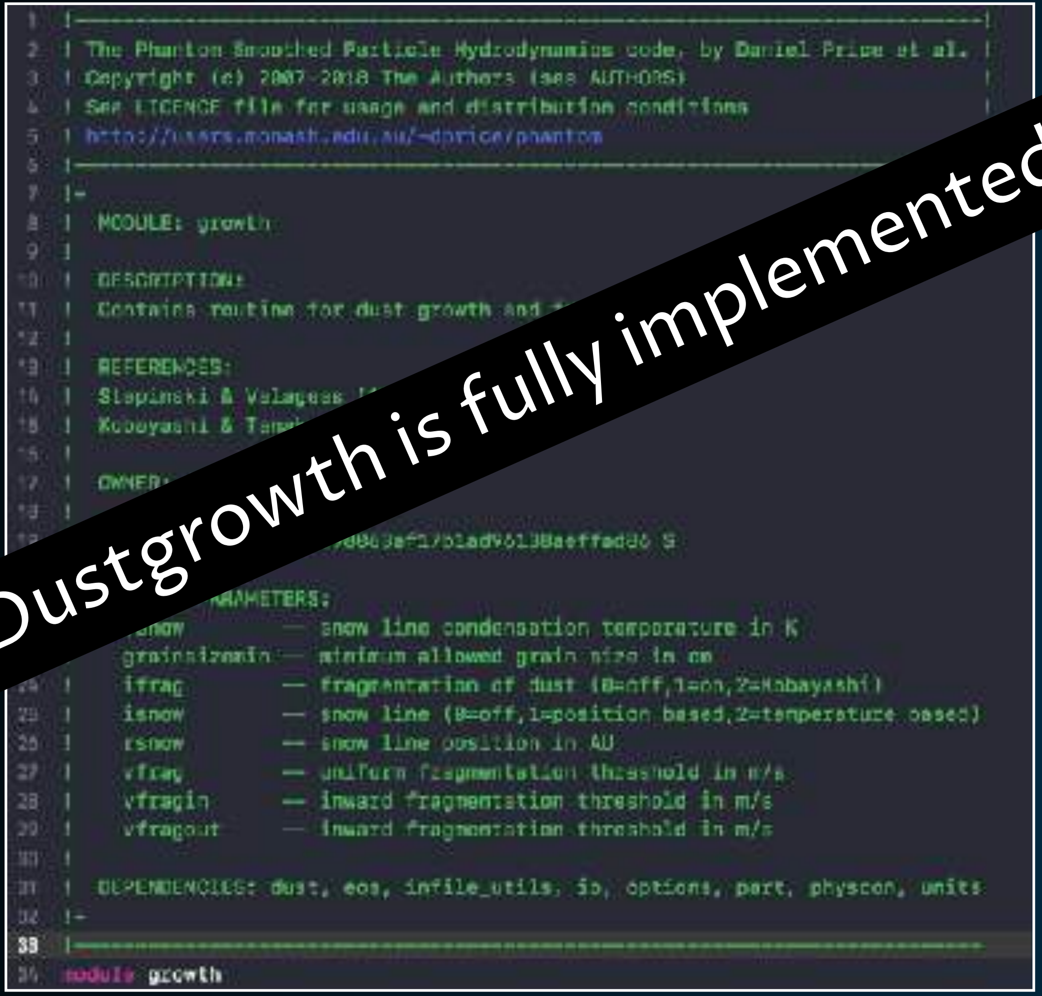


Dustgrowth is fully implemented

```
!--------------------------------------------------------------------------!
! The Phantom Smoothed Particle Hydrodynamics code, by Daniel Price et al. !
! Copyright (c) 2007-2018 The Authors (see AUTHORS)                        !
! See LICENCE file for usage and distribution conditions                   !
! http://users.monash.edu.au/~dprice/phantom                               !
!--------------------------------------------------------------------------!
!+
!  MODULE: growth
!
!  DESCRIPTION:
!  Contains routine for dust growth and
!
!  REFERENCES:
!  Stepinski & Valageas
!  Kobayashi & Tanak
!
!  OWNER:
!
                     0863af17b1ad96138aeffadd6 $

             ARAMETERS:
      snow        -- snow line condensation temperature in K
    grainsizemin -- minimum allowed grain size in cm
!    ifrag        -- fragmentation of dust (0=off,1=on,2=Kobayashi)
!    isnow        -- snow line (0=off,1=position based,2=temperature based)
!    rsnow        -- snow line position in AU
!    vfrag        -- uniform fragmentation threshold in m/s
!    vfragin      -- inward fragmentation threshold in m/s
!    vfragout     -- inward fragmentation threshold in m/s
!
!  DEPENDENCIES: dust, eos, infile_utils, io, options, part, physcon, units
!+
!--------------------------------------------------------------------------
module growth
```

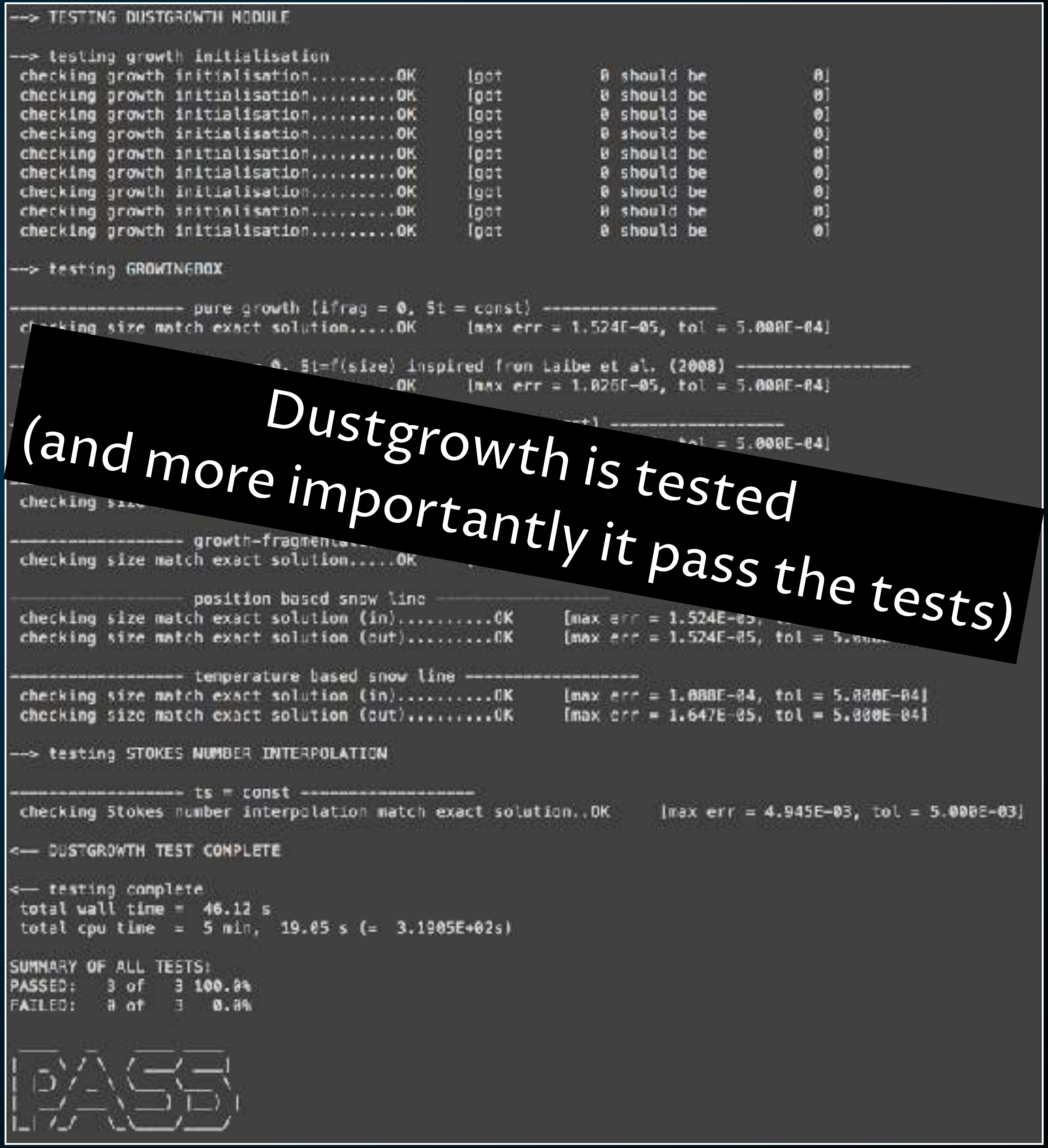
CONCLUSION

100 %



```
!--------------------------------------------------------------------------!
! The Phantom Smoothed Particle Hydrodynamics code, by Daniel Price et al. !
! Copyright (c) 2007-2018 The Authors (see AUTHORS)                        !
! See LICENCE file for usage and distribution conditions                   !
! http://users.monash.edu.au/~dprice/phantom                               !
!--------------------------------------------------------------------------!
!+
!  MODULE: growth
!
!  DESCRIPTION:
!  Contains routine for dust growth and
!
!  REFERENCES:
!  Stepinski & Valageas
!  Kobayashi & Tana
!
!  OWNER:
!
                    0863af17b1ad9613Baeffadd6 $

          RAMETERS:
     snow          -- snow line condensation temperature in K
    grainsizemin  -- minimum allowed grain size in cm
!    ifrag         -- fragmentation of dust (0=off,1=on,2=Kobayashi)
!    isnow         -- snow line (0=off,1=position based,2=temperature based)
!    rsnow         -- snow line position in AU
!    vfrag         -- uniform fragmentation threshold in m/s
!    vfragin       -- inward fragmentation threshold in m/s
!    vfragout      -- inward fragmentation threshold in m/s
!
!  DEPENDENCIES: dust, eos, infile_utils, io, options, part, physcon, units
!+
!--------------------------------------------------------------------------
module growth
```

Dustgrowth is fully implemented

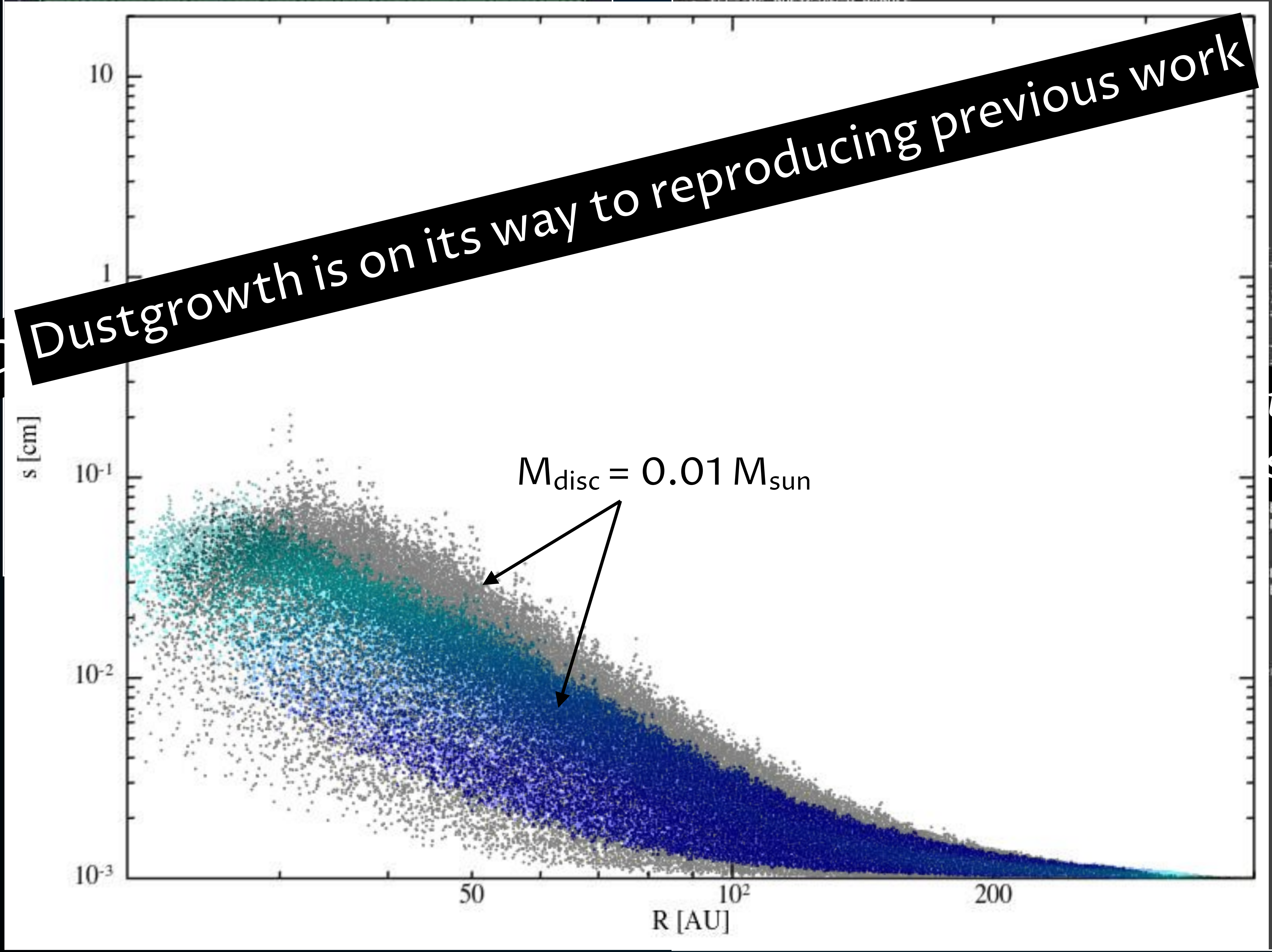


```
--> TESTING DUSTGROWTH MODULE

--> testing growth initialisation
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]
 checking growth initialisation.........OK     [got          0 should be          0]

--> testing GROWINGBOX

------------------ pure growth (ifrag = 0, St = const) ------------------
 checking size match exact solution.....OK     [max err = 1.524E-05, tol = 5.000E-04]

                 0, St=f(size) inspired from Laibe et al. (2008) ------------------
                                        OK     [max err = 1.026E-05, tol = 5.000E-04]

                                        t) ------------------
                                                          tol = 5.000E-04]

 checking size

------------------ growth-fragmenta
 checking size match exact solution.....OK

------------------ position based snow line ------------------
 checking size match exact solution (in)..........OK     [max err = 1.524E-05,
 checking size match exact solution (out).........OK     [max err = 1.524E-05, tol = 5.

------------------ temperature based snow line ------------------
 checking size match exact solution (in)..........OK     [max err = 1.088E-04, tol = 5.000E-04]
 checking size match exact solution (out).........OK     [max err = 1.647E-05, tol = 5.000E-04]

--> testing STOKES NUMBER INTERPOLATION

------------------ ts = const ------------------
 checking Stokes number interpolation match exact solution..OK     [max err = 4.945E-03, tol = 5.000E-03]

<-- DUSTGROWTH TEST COMPLETE

<-- testing complete
 total wall time =   46.12 s
 total cpu time  =   5 min,  19.05 s (=  3.1905E+02s)

SUMMARY OF ALL TESTS:
PASSED:    3 of    3 100.0%
FAILED:    0 of    3   0.0%

 ____   _    ____ ____
|  _ \ / \  / ___/ ___|
| |_) / _ \ \___ \___ \
|  __/ ___ \ ___) |__) |
|_| /_/   \_\____/____/
```

Dustgrowth is tested
(and more importantly it pass the tests)

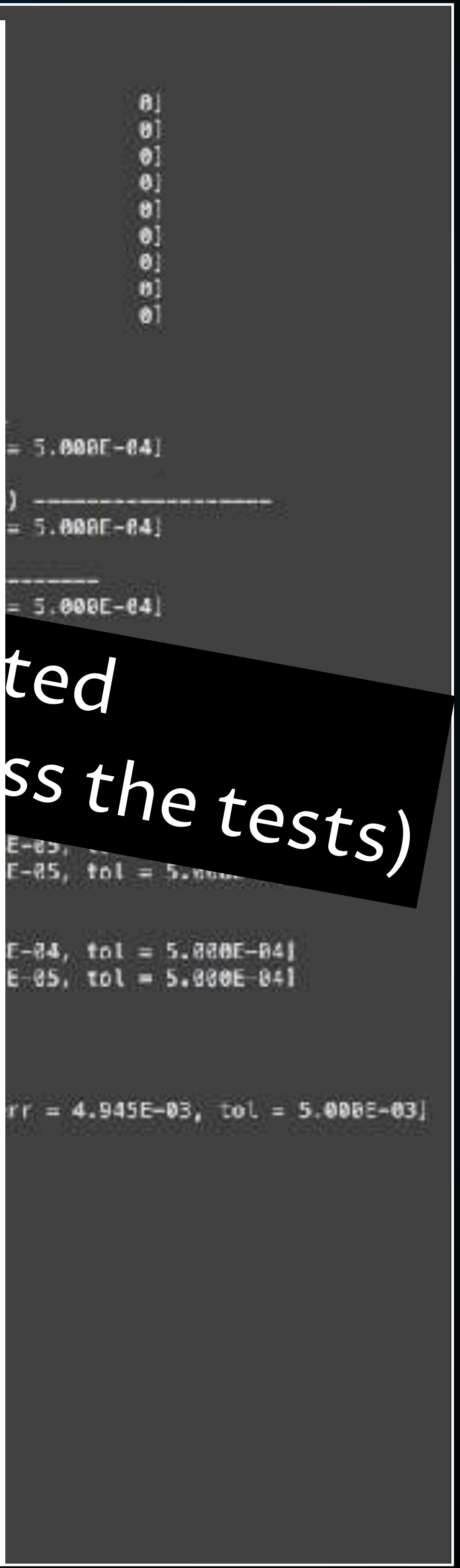
CONCLUSION

100 %



$M_{disc} = 0.01\,M_{sun}$

Dustgrowth is on its way to reproducing previous work



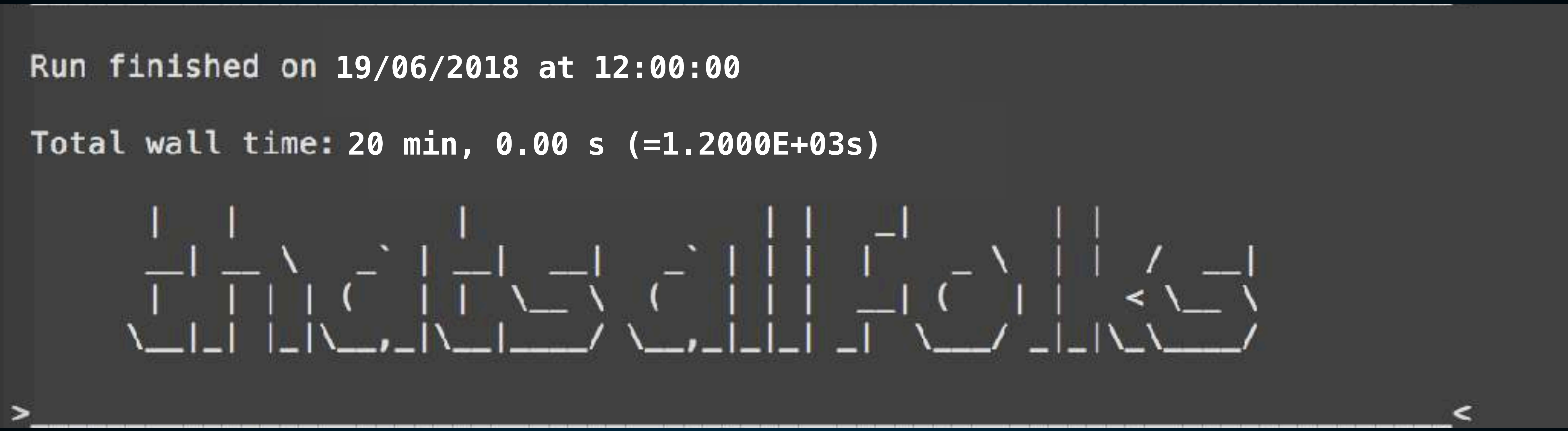
```
                            0]
                            0]
                            0]
                            0]
                            0]
                            0]
                            0]
                            0]
                            0]

= 5.000E-04]

) -----------------
= 5.000E-04]

--------
= 5.000E-04]

E-05,
E-05, tol = 5.

E-04, tol = 5.000E-04]
E-05, tol = 5.000E-04]

rr = 4.945E-03, tol = 5.000E-03]
```

D

ted

ss the tests)

THANK YOU



```
Run finished on 19/06/2018 at 12:00:00

Total wall time: 20 min, 0.00 s (=1.2000E+03s)
 _   _           _              _ _    __       _ _        
| |_| |__   __ _| |_ ___   __ _| | |  / _| ___ | | | _____ 
| __| '_ \ / _` | __/ __| / _` | | | | |_ / _ \| | |/ / __|
| |_| | | | (_| | |_\__ \| (_| | | | |  _| (_) | |   <\__ \
 \__|_| |_|\__,_|\__|___/ \__,_|_|_| |_|  \___/|_|_|\_\___/

>                                                          <
```

CRAL

CENTRE DE RECHERCHE ASTROPHYSIQUE DE LYON

observatoire de Lyon

PHANTOM