The Great Merge

Progress towards a unified RAMSES

Noé Brucy
Benoît Commercon
Matthias Gonzalez
Patrick Hennebelle
Ugo Lebreuilly
Romain Teyssier
Antoine Verliat

Tine Colman

Motivation

RAMSES has a large variety of users

Many versions of the code exist

Possible issues:

- Bug fixes are not always applied to every version
- Duplicated work
- Incompatible physics features
- Complicates comparison of results

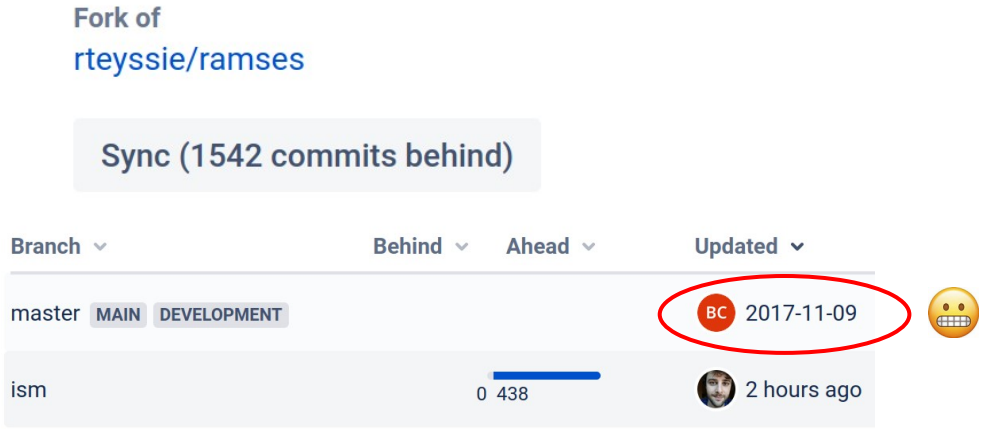
SOLUTION: Merge everything back into 1 central repository

Advantages:

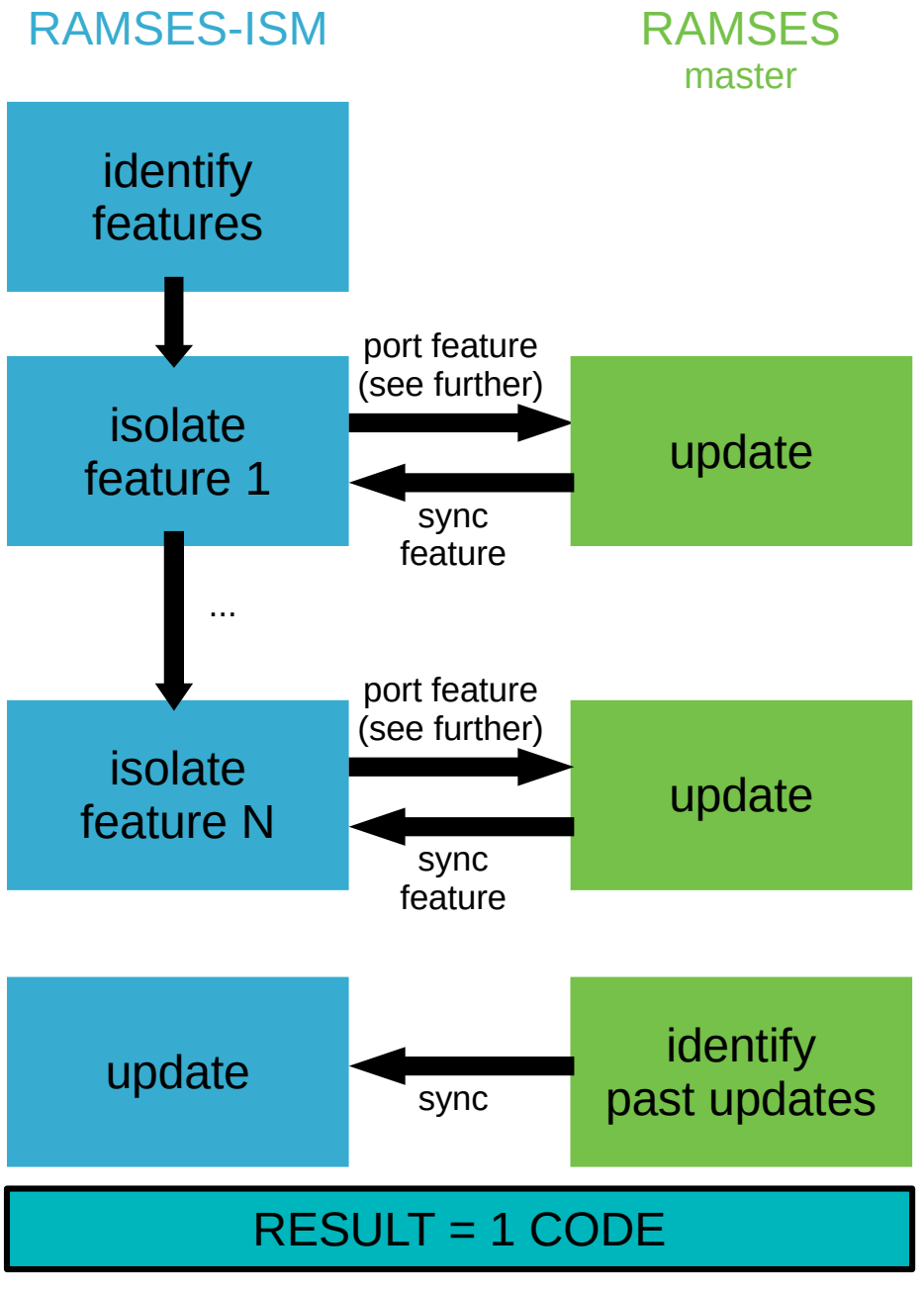
- Overview of past/ongoing development
- More available physics
- Code reviews by other users
- Less bugs!
- Easier maintenance
- ...

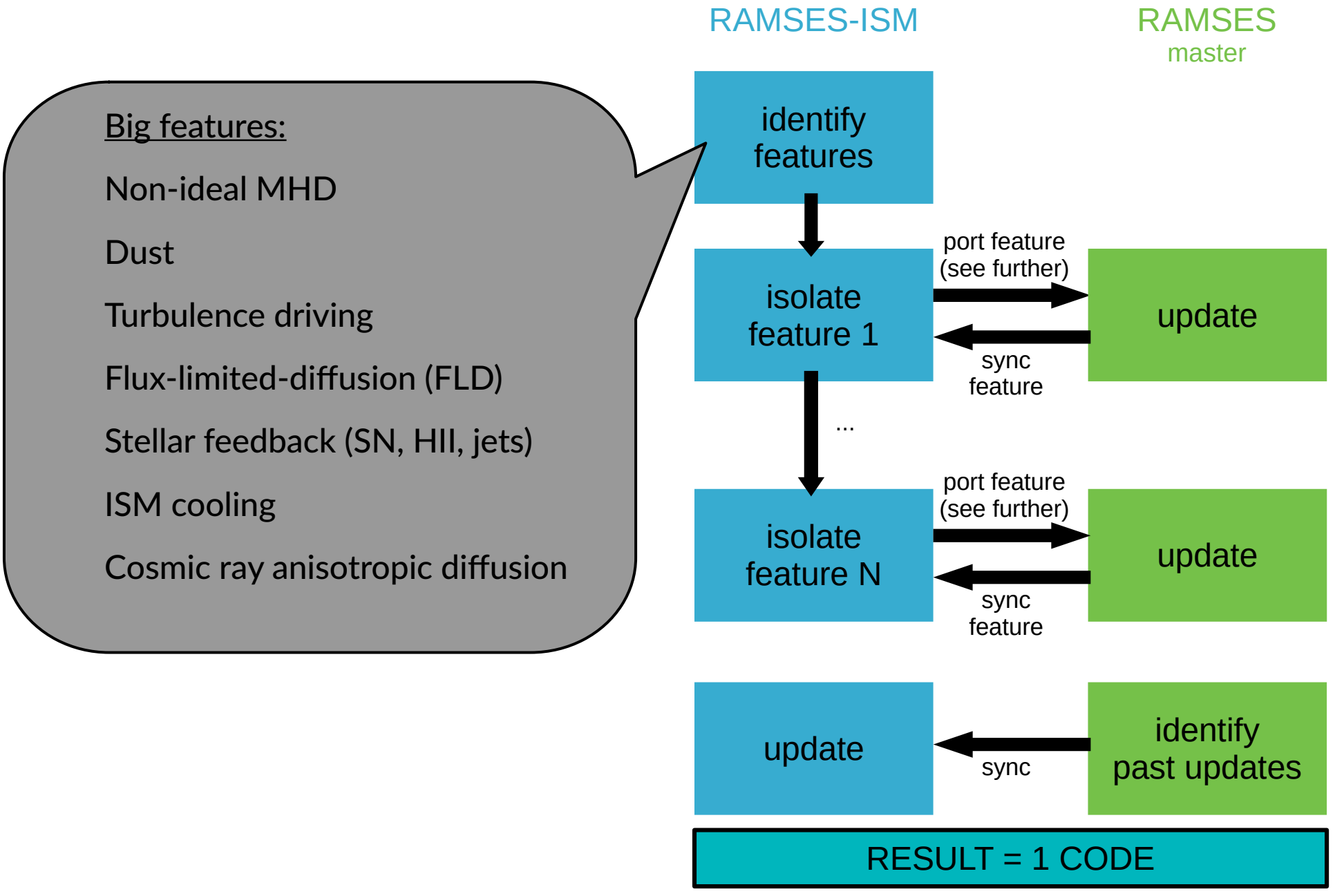
Merging process

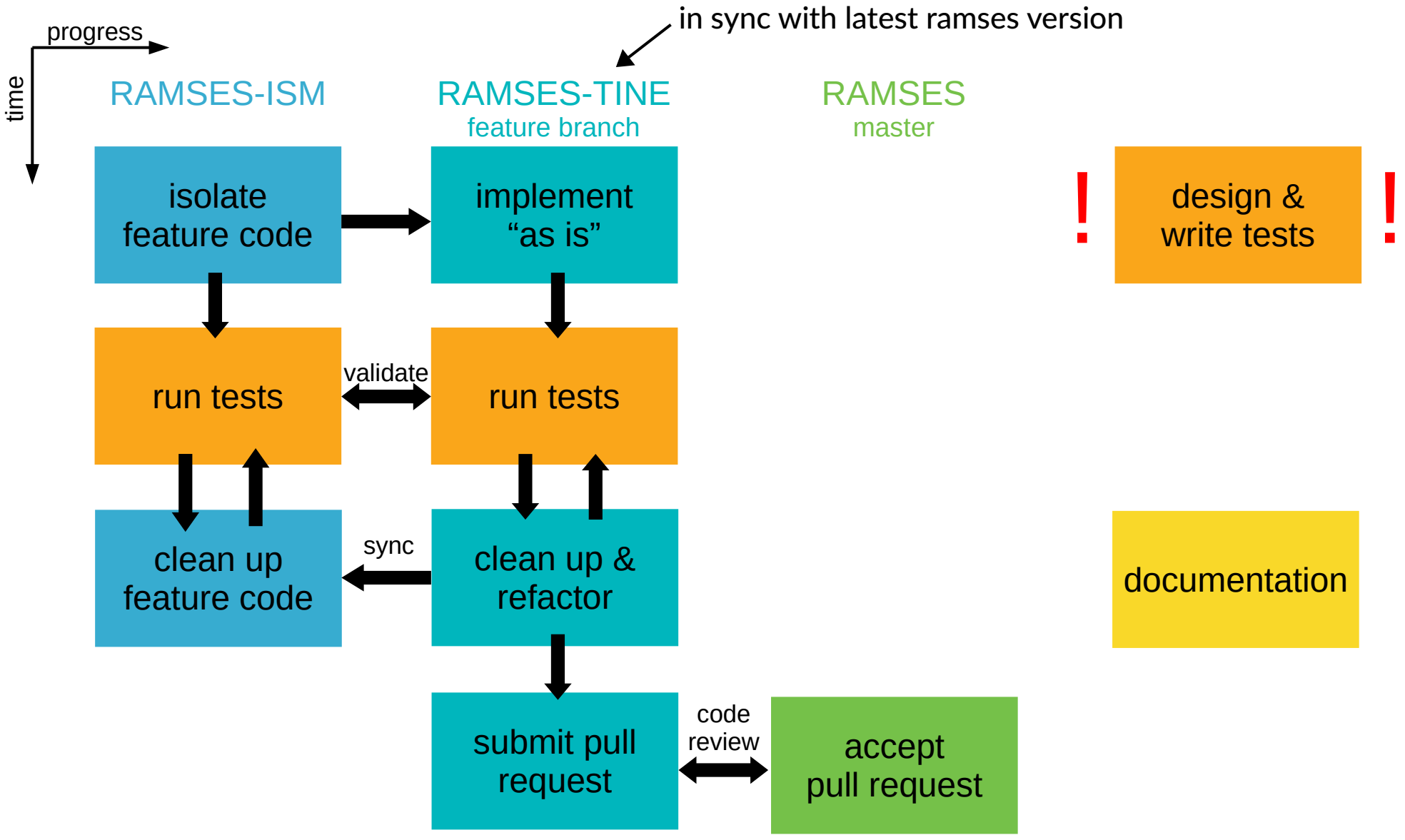
The case of ramses-ism

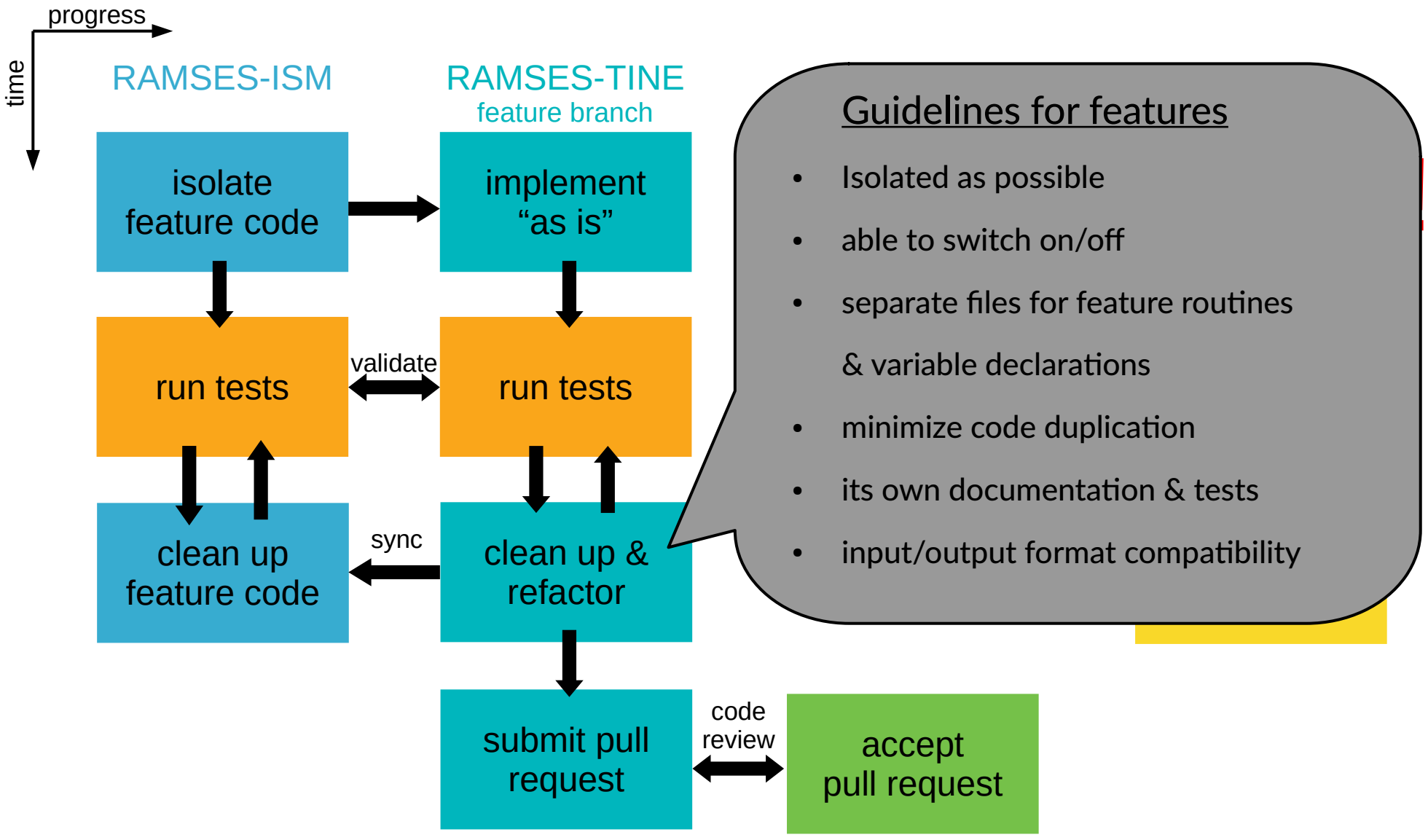


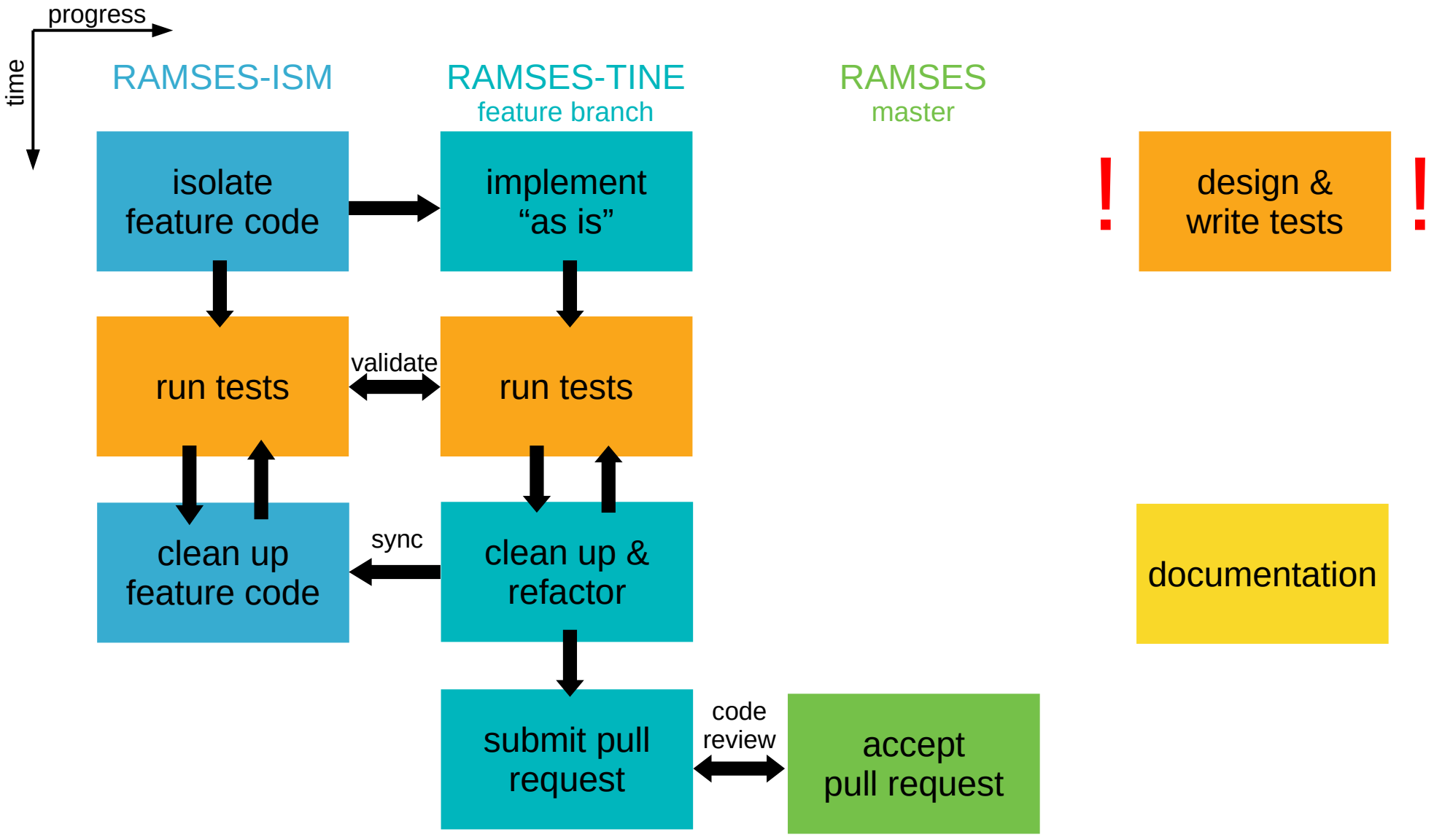
Not possible to do a normal merge


RAMSES-ISM
RAMSES
master
identify features
isolate feature 1
port feature (see further)
sync feature
update
...
isolate feature N
port feature (see further)
sync feature
update
update
sync
identify past updates
RESULT = 1 CODE


RAMSES-ISM
RAMSES
master
Big features:
Non-ideal MHD
Dust
Turbulence driving
Flux-limited-diffusion (FLD)
Stellar feedback (SN, HII, jets)
ISM cooling
Cosmic ray anisotropic diffusion
identify features
isolate feature 1
port feature (see further)
sync feature
update
...
isolate feature N
port feature (see further)
sync feature
update
update
sync
identify past updates
RESULT = 1 CODE

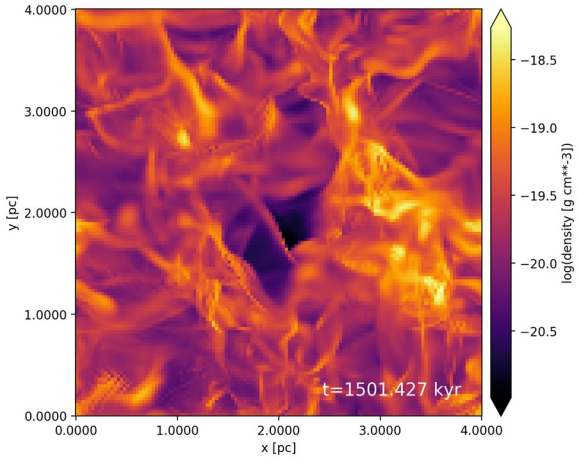

progress
time
in sync with latest ramses version
RAMSES-ISM
RAMSES-TINE
feature branch
RAMSES
master
isolate feature code
implement "as is"
design & write tests
!
!
run tests
validate
run tests
clean up feature code
sync
clean up & refactor
documentation
submit pull request
code review
accept pull request


progress
time
RAMSES-ISM
RAMSES-TINE
feature branch
isolate feature code
implement "as is"
run tests
validate
run tests
clean up feature code
sync
clean up & refactor
submit pull request
code review
accept pull request
Guidelines for features
Isolated as possible
able to switch on/off
separate files for feature routines & variable declarations
minimize code duplication
its own documentation & tests
input/output format compatibility


progress
time
RAMSES-ISM
RAMSES-TINE
feature branch
RAMSES
master
isolate feature code
implement "as is"
! design & write tests !
run tests
validate
run tests
clean up feature code
sync
clean up & refactor
documentation
submit pull request
code review
accept pull request

Example: Turbulence driving

- Adds forcing to box to generate/maintain turbulence
- Andrew McLeod, in 2014
- Was easy because standalone module
- Dependency: fftw3
- Implementation:



generalised Ornstein-Uhlenbeck process

random growth decay

$$\mathrm{d}F(k,t) = F_0(k)P(k)\mathrm{d}W_t - F(k,t)\mathrm{d}t/T$$

auto-correlation time

Wiener process

projection operator

power spectrum

amr/amr_step.f90 MODIFIED

```
...
10  10    use coolrates_module, only: update_coolrates_tables
11  11    use rt_cooling_module, only: update_UVrates
12  12   #endif
    13   +#if USE_TURB==1
    14   +  use turb_commons
    15   +#endif
13  16    use mpi_mod
14  17    implicit none
15  18   #ifndef WITHOUTMPI
...
286 289    if(rt .and. rt_star) call update_star_RT_feedback(ilevel)
287 290   #endif
288 291
    292   +#if USE_TURB==1
    293   +  ! Compute turbulent forcing
    294   +                              call timer('turb','start')
    295   +  if (turb .and. turb_type/=3) then
    296   +     ! Calculate turbulent acceleration on each cell in this level
    297   +     call calc_turb_forcing(ilevel)
    298   +  end if
    299   +#endif
    300   +
289 301    !----------------------
290 302    ! Compute new time step
291 303    !----------------------
...
384 396                               call timer('poisson','start')
385 397       if(poisson)call synchro_hydro_fine(ilevel,+0.5*dtnew(ilevel))
386 398
    399   +#if USE_TURB==1
    400   +     ! Compute turbulent forcing
    401   +                              call timer('turb','start')
    402   +     if (turb .AND. turb_type/=3) then
    403   +        ! Euler step, adding turbulent acceleration
    404   +        call add_turb_forcing(ilevel,dtnew(ilevel))
    405   +     end if
    406   +
    407   +#endif
    408   +
387 409       ! Restriction operator
388 410                               call timer('hydro upload fine','start')
389 411       call upload_fine(ilevel)
...
```

preprocessor directive

Files changed (23)

+6	-0	M	amr/adaptive_loop.f90
+22	-0	M	amr/amr_step.f90
+14	-0	M	amr/output_amr.f90
+3	-0	M	amr/read_params.f90
+16	-0	M	amr/update_time.f90
+22	-1	M	bin/Makefile
+13	-0	M	hydro/courant_fine.f90
+11	-0	M	hydro/init_flow_fine.f90
+13	-0	M	mhd/courant_fine.f90
+11	-0	M	mhd/init_flow_fine.f90
+10	-0	M	pm/newdt_fine.f90
+11	-0	A	turb/README
+216	-0	A	turb/add_turb_forcing.f90
+215	-0	A	turb/init_turb.f90
+47	-0	A	turb/mpi_share_turb_fields.f90
+57	-0	A	turb/read_turb_fields.f90
+58	-0	A	turb/read_turb_params.f90
+44	-0	A	turb/turb_check_time.f90
+31	-0	A	turb/turb_commons.f90
+770	-0	A	turb/turb_force_utils.f90
+35	-0	A	turb/turb_next_field.f90
+54	-0	A	turb/turb_parameters.f90
+59	-0	A	turb/write_turb_fields.f90

minimal changes to existing files

New directory for feature

Makefile:

```
# Use turbulence? 1=Yes, 0=No (requires fftw3)
USE_TURB=1
```

```
ifeq ($(USE_TURB),1)
   LIBS_TURB = -L$(HOME)/local/lib -lfftw3
   LIBS_OBJ_TURB = -I$(HOME)/local/include -lfftw3
endif
```

Namelist:

```
&TURB_PARAMS
turb=.true.
turb_type=1          ! 1=forced evolving, 2=forced fix
turb_seed=1          ! -1 = random
comp_frac=0.33       ! compressive fraction
instant_turb=.true. ! Generate initial turbulence bef
forcing_power_spectrum='parabolic' !powerlaw, parabol
turb_T=0.001         ! Turbulent velocity autocorrelati
turb_Ndt=100         ! Number of timesteps per autocorr
turb_rms=1.e4        ! rms turbulent forcing accelerati
turb_min_rho=1d-50 ! Minimum density for turbulence
/
```

Wiki

Create pa

ramses / **TurbulenceDriving**

Turbulence driving

The block named `&TURB_PARAMS` contains the parameters related to the turbulence driving. Originally implemented by Andrew Mcleod

Overview of parameters

Variable name	Fortran type	Default value	Description
`turb`	`boolean`	`.false.`	Turn on or off driving
`turb_seed`	`integer`	`-1`	Random number generator seed. -1 = random
`turb_type`	`integer`	`1`	How the driving changes over time. 1=driven evolving, 3=decaying
`instant_turb`	`boolean`	`.true.`	Generate initial turbulence before start
`comp_frac`	`float`	`0.3333`	The weight of compressive over solenoidal modes
`turb_T`	`float`	`1`	Turbulent velocity auto-correlation time in code units.
`turb_Ndt`	`integer`	`100`	Number of timesteps per auto-correlation time
`turb_rms`	`float`	`1`	Root-mean-square turbulent forcing in code units.
`turb_min_rho`	`float`	`1d-50`	Minimum density for turbulence. Not forcing is added onto cellswith a density less than this value.
`forcing_power_spectrum`	`string`	`parabolic`	Power spectrum type of the forcing, which describes the relative strength of individual modes. Options are: power_law, parabolic, konstandin

Test case

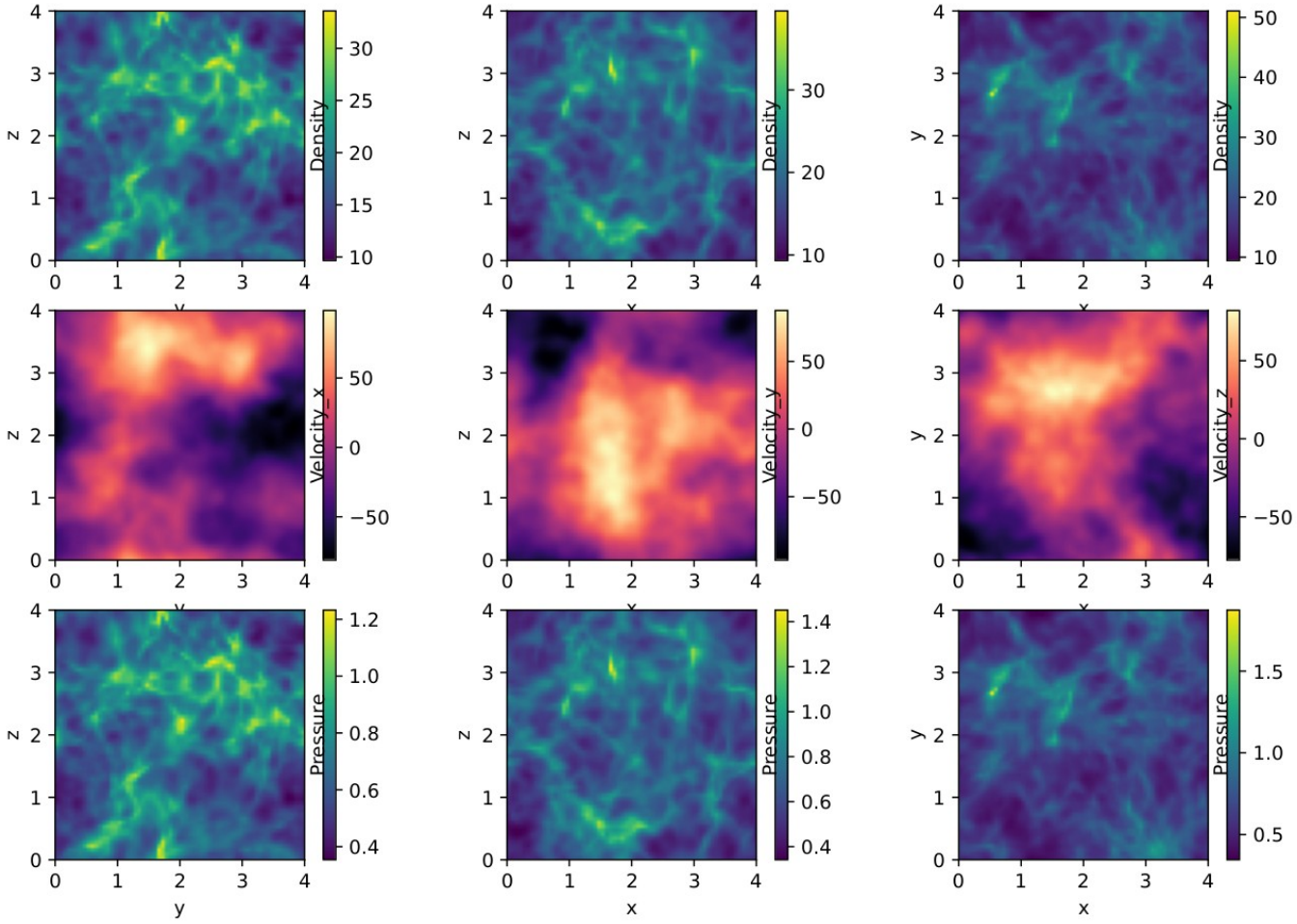
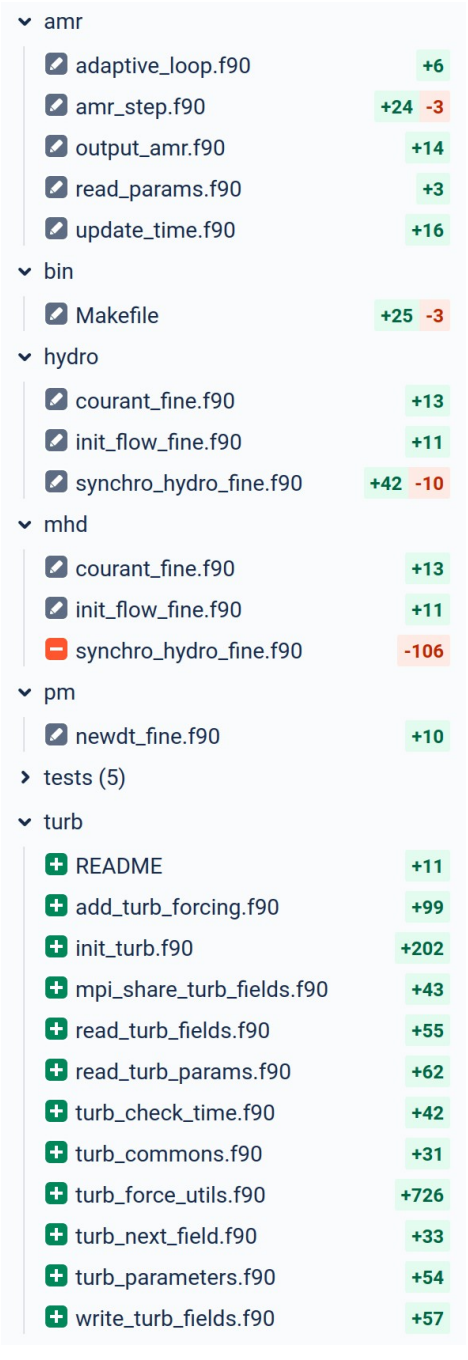


Figure 1: turb/driving test



Final PR

Turbulence driving update

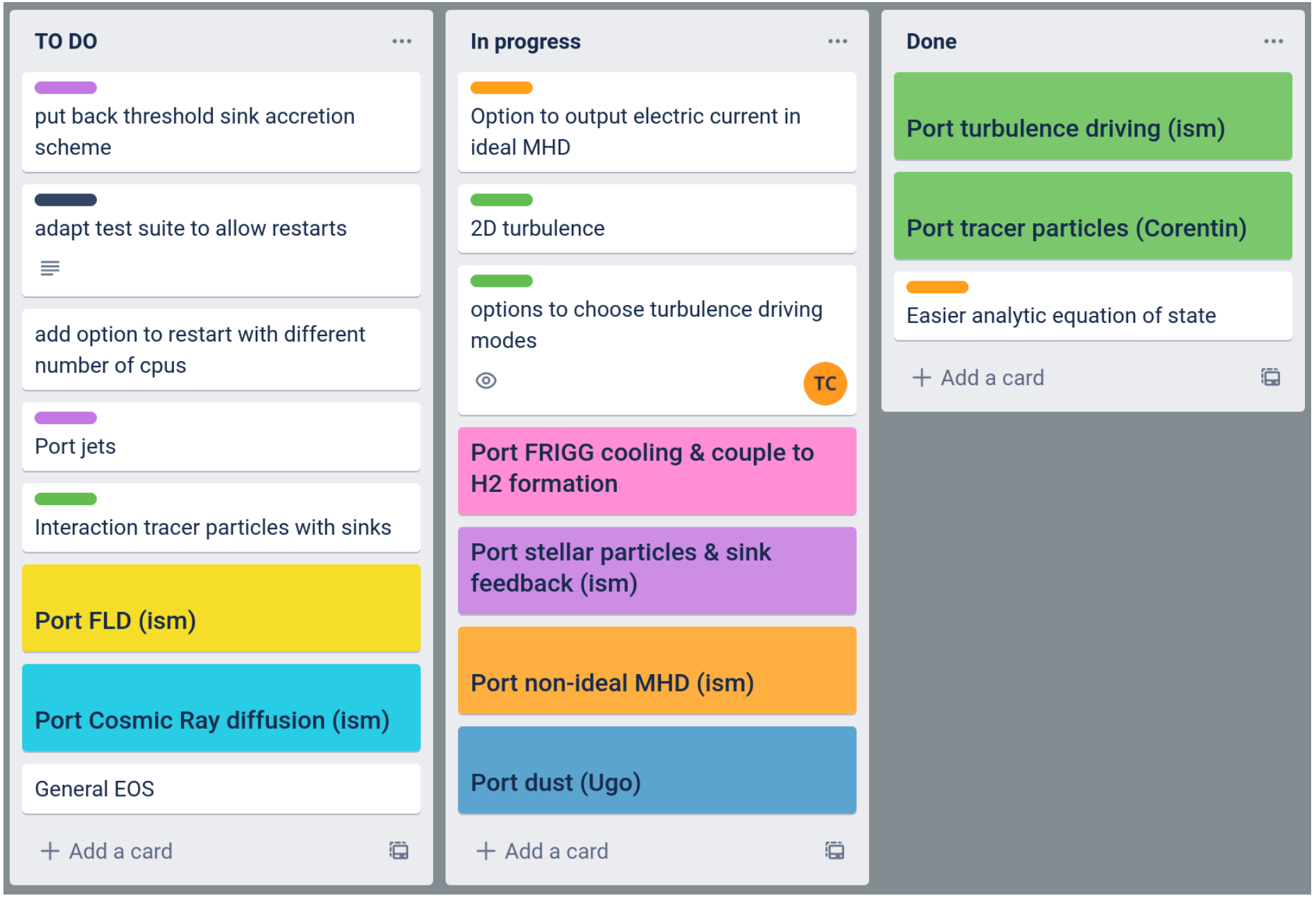
https://bitbucket.org/rteyssie/ramses/pull-requests/447

Now available in main ramses!

[WIP] More options to select modes

[WIP] User guide

- description implementation
- what parameters to choose


TO DO
put back threshold sink accretion scheme
adapt test suite to allow restarts
add option to restart with different number of cpus
Port jets
Interaction tracer particles with sinks
Port FLD (ism)
Port Cosmic Ray diffusion (ism)
General EOS
Add a card
In progress
Option to output electric current in ideal MHD
2D turbulence
options to choose turbulence driving modes
TC
Port FRIGG cooling & couple to H2 formation
Port stellar particles & sink feedback (ism)
Port non-ideal MHD (ism)
Port dust (Ugo)
Add a card
Done
Port turbulence driving (ism)
Port tracer particles (Corentin)
Easier analytic equation of state
Add a card

TO DO

Masson et al 2012
Marchand et al 2018

Ambipolar diffusion ✓

Ohmic dissipation ✓

Hall effect ✗

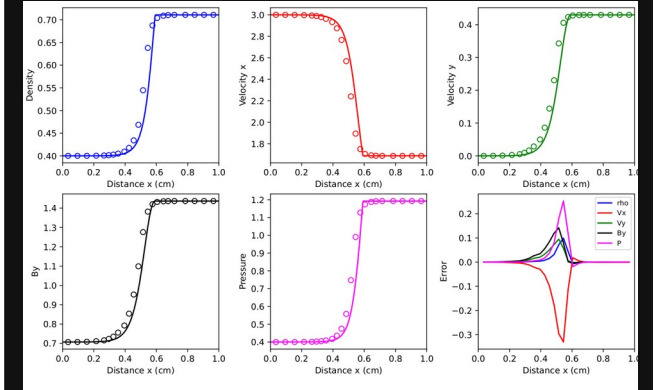


Figure 2: nimhd/nimhd-cshock-ohm test

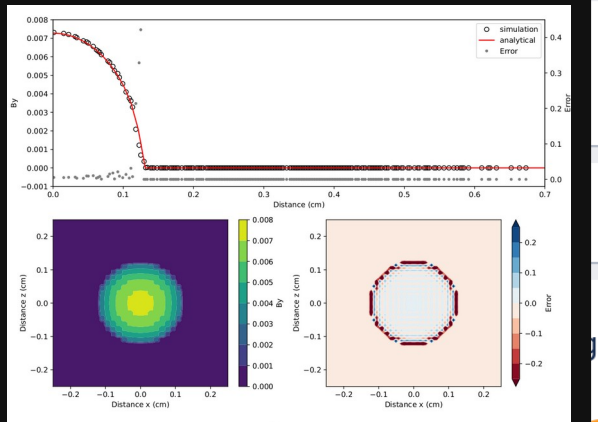


Figure 3: nimhd/nimhd-diffusion-ad test

Table 3: nimhd-diffusion-ad error summary

Variable	This run	Reference	Error	Tolerance
B x left	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
B x right	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
B y left	6.399999999993790e+01	6.400000000000014e+01	9.725553695717315e-14	2.999999999999998e-13
B y right	6.399999999993790e+01	6.400000000000014e+01	9.725553695717315e-14	2.999999999999998e-13
B z left	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
B z right	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
boden	1.000000000000000e+00	1.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
current x	7.586446588798568e+02	7.592928512366497e+02	8.544083836924322e-04	2.999999999999998e-13
current y	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
current z	7.586446588798568e+02	7.592928512366497e+02	8.544083836924322e-04	2.999999999999998e-13
density	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
dx	1.213000000000000e+03	1.213000000000000e+03	0.000000000000000e+00	2.999999999999998e-13
level	3.401280000000000e+05	3.401280000000000e+05	0.000000000000000e+00	2.999999999999998e-13
ncells	5.942400000000000e+04	5.942400000000000e+04	0.000000000000000e+00	2.999999999999998e-13
pressure	-2.364718684206752 4e+04	0.000000000000000e+00	inf	2.999999999999998e-13
time	5.040249667819310 3e+01	5.021531169574400e+01	3.803220180117077 0e-03	2.999999999999998e-13
unit d	1.000000000000000e+00	1.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
unit l	1.000000000000000e+00	1.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
unit t	1.000000000000000e+00	1.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
velocity x	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
velocity y	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
velocity z	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	2.999999999999998e-13
x	2.982750000000000e+04	2.982750000000000e+04	0.000000000000000e+00	2.999999999999998e-13
y	2.971200000000000e+04	2.971200000000000e+04	0.000000000000000e+00	2.999999999999998e-13
z	2.982750000000000e+04	2.982750000000000e+04	0.000000000000000e+00	2.999999999999998e-13

Port Cosmic Ray diffusion (ism)

General EOS

+ Add a card

In progress

Port non-ideal MHD (ism)

Port dust (Ugo)

+ Add a card

Done

Port turbulence driving (ism)

Port tracer particles (Corentin)

Easier analytic equation of state

+ Add a card

TO DO

put back threshold sink accretion

In progress

Option to output electric current in

Done

Port turbulence driving (ism)

Port tracer particles (Corentin)

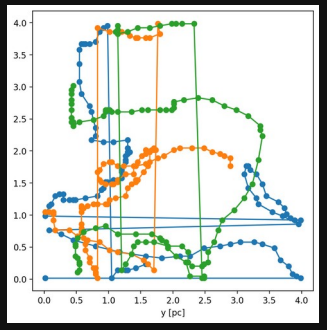
Easier analytic equation of state

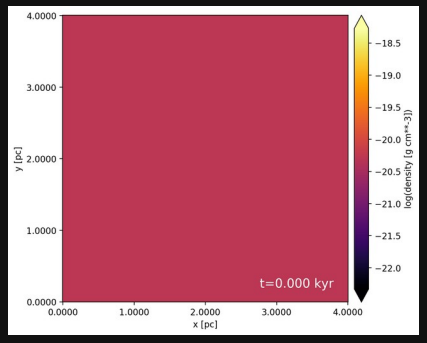
+ Add a card

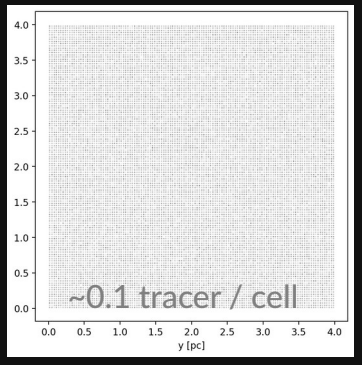
Cadiou et al 2019

```
&TRACER_PARAMS
tracer=.true.
MC_tracer=.true.
tracer_feed_fmt='inplace'   !can also initialize from file
tracer_mass=0.00001 !this gives about 1 particle per cell
/
```

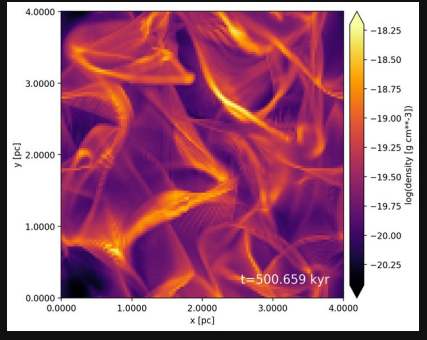
Example trajectories

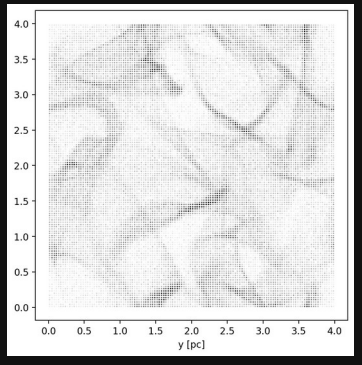


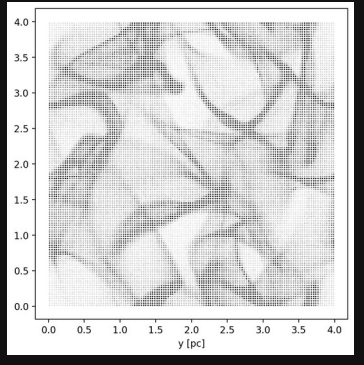


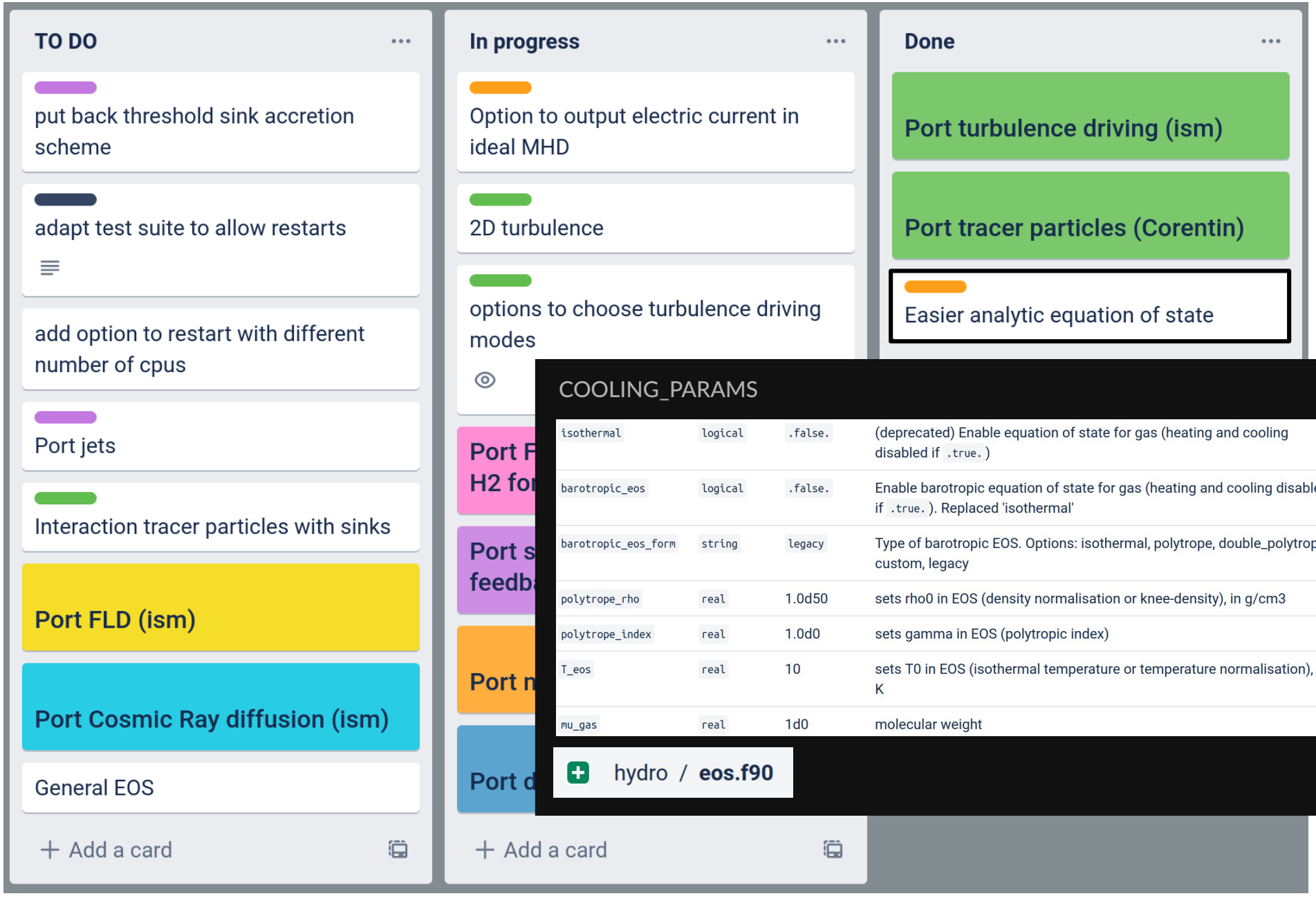


~1 tracer / cell








TO DO
put back threshold sink accretion scheme
adapt test suite to allow restarts
add option to restart with different number of cpus
Port jets
Interaction tracer particles with sinks
Port FLD (ism)
Port Cosmic Ray diffusion (ism)
General EOS
+ Add a card
In progress
Option to output electric current in ideal MHD
2D turbulence
options to choose turbulence driving modes
Port F
H2 for
Port s
feedb
Port n
Port d
+ Add a card
Done
Port turbulence driving (ism)
Port tracer particles (Corentin)
Easier analytic equation of state
COOLING_PARAMS
isothermal logical .false. (deprecated) Enable equation of state for gas (heating and cooling disabled if .true.)
barotropic_eos logical .false. Enable barotropic equation of state for gas (heating and cooling disabled if .true.). Replaced 'isothermal'
barotropic_eos_form string legacy Type of barotropic EOS. Options: isothermal, polytrope, double_polytrope, custom, legacy
polytrope_rho real 1.0d50 sets rho0 in EOS (density normalisation or knee-density), in g/cm3
polytrope_index real 1.0d0 sets gamma in EOS (polytropic index)
T_eos real 10 sets T0 in EOS (isothermal temperature or temperature normalisation), in K
mu_gas real 1d0 molecular weight
hydro / eos.f90

How can you help?

- Use git branches instead of patches for development
- Add your typical (low resolution) setup as a test case
- Merge your features into the main repo

Join RAMSES merge slack (or make a specific feature channel on the teamramses slack)

https://join.slack.com/t/mergingramsesversions/shared_invite/zt-wat3j23a-~Q01jBM~9x7douyYJOX2wA