0.1 Outline of the parallel implementation

A significant amount of effort has been put in to make the whole modelling capability parallelisable. Indeed, while the code comes with a 0-halo parallelisation method, the correct operation of the high-order WENO scheme (see Chapter **??**) requires a n-halo approach.

0-halo parallelisation of OpenFOAM

The 0-halo approach implemented in OpenFOAM consists in dividing the global domain D in non-overlapping sub-domains D_i (see Figure **??**). The transfer of information from one sub-domain to the other is therefore performed through the sub-domain boundaries only. It indicates that the numerical schemes used to discretise the governing equations become at best second-order accurate at the inter-processor boundaries.

Although this methodology is appropriate for the solution of the incompressible Navier-Stokes equations discretised with second-order schemes, the 0-halo parallel transport of the level set, using a high-order scheme, leads to non-physical behaviour of the interface at the inter-processor boundaries.

n-halo parallel transport of the level set

The n-halo parallelisation of the RCLS alone was first performed by simply producing a n-halo decomposition of the domain (see Figure 1) and updating in each extended sub-domain Dh_i, the halo cells according to the solution on the neighbouring sub-domains Dh_j. The pre-processing variables are then produced for the extended sub-domains such that the WENO scheme performs with the sought high-order accuracy on the non-extended part of Dh_i.

The halos are produced by gathering layers of cells around the non-extended sub-domain D_i. The number of layers N_l is a function of the order of the scheme and is calculated such that the solution in the first layer of halo cells L_1, is high-order

accurate. This ensures that the flux exchanged between D_i and L_1 is high-order accurate, such that the solution is properly calculated in all the cells of the non-extended part of Dh_i.

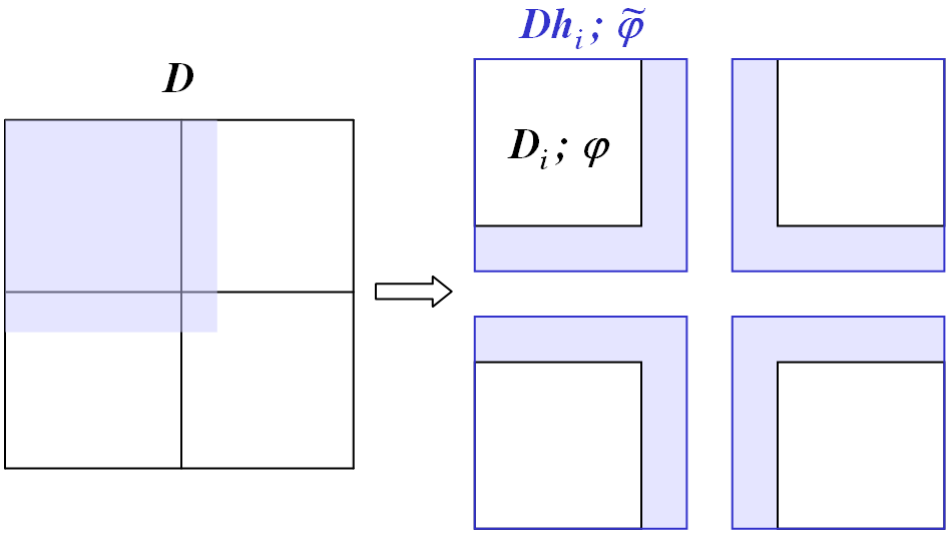

Figure 1: Decomposition of the domain using a n-halo approach — On the left the global domain (0-halo inter-processor boundaries in black; extent of a n-halo sub-domain in grey); on the right the 0-halo sub-domains D_i (black line), the extended sub-domains Dh_i (blue line) and the halos (grey region)

However, the modelling capability involves not only the transport of the level set but also the solution of the pressure-velocity coupling. As the incompressible Navier-Stokes equations are discretised semi-implicitly (see Section **??**), the resulting system of algebraic equations (see Section **??**) is solved by linear solvers. These linear solvers are implemented in OpenFOAM for a 0-halo parallelisation and can only run on a domain decomposed into non-overlapping domains.

n-halo parallel simulation of two-phase flows

An extension of the above n-halo methodology has been produced to transport the level set with high-order accuracy and solve, in the same computation, the in-

compressible Navier-Stokes equations with the linear solvers implemented in OpenFOAM.

Because of these linear solvers, the calculation has to run on a domain decomposed with a 0-halo approach. In order to maintain the high-order accuracy of the level set transport, the pre-processing variables are calculated on the extended sub-domains and re-written for the 0-halo sub-domains. This re-writing process involves the creation of D_i-to-Dh_i "maps" that relate the geometrical features of the 0-halo sub-domains to their counterparts in the extended sub-domains Dh_i. It also implies generating new lists to store the mesh-dependent variables of the first layer of halo cells L_1. These variables are indeed required by the WENO scheme to perform high-order calculations of the flux at the inter-processor boundaries.

At run time, the level set transport is performed on the 0-halo sub-domains. However, as the re-written mesh-dependent variables refer to cells in the extended sub-domains, an intermediate "virtual field" $\tilde{\varphi}$ is used to perform the operations of the WENO scheme. This field is of the size of the extended sub-domains Dh_i and is populated with the level set scalar. The halo cells in $\tilde{\varphi}$ are updated, through MPI transfers, according to the level set field in the neighbouring 0-halo sub-domains. The level set field φ is then calculated in the 0-halo sub-domain from the virtual field $\tilde{\varphi}$.

As implemented in the code, the n-halo parallelisation involves the following steps:

1. Decomposition of the global domain D into extended sub-domains Dh_i.

2. Calculation of the mesh-dependent variables on the extended sub-domains. At this step, both these pre-processing variables and some geometrical characteristics of Dh_i are written out to construct the D_i-to-Dh_i maps at a later stage.

3. Decomposition of the global domain D into 0-halo sub-domains D_i.

4. Construction of the D_i-to-Dh_i maps that relate the 0-halo sub-domains to the extended sub-domains.

5. Re-writing of the pre-processing variables for D_i using the D_i-to-Dh_i maps.

6. Parallel run of the computation using a virtual field $\tilde{\varphi}$ to transport the level set field φ with high-order accuracy.

Droplet transfer in parallel

Just like the rest of the modelling capability, the transfer of droplets has been implemented in parallel. The parallel run of the algorithm on several processors involves two specific cases: droplets spanning on several sub-domains and the re-initialisation of the level set field φ on all cores simultaneously if a transfer of droplet has occurred somewhere in the domain.

As mentioned in Section **??**, droplets spanning on several processors are ignored by the selection algorithm. Therefore, these drops remain in the level set formulation until they are convected well inside a sub-domain. The test that handles this case checks that none of the cells of the droplet considered are on a processor boundary. This test is made computationally efficient by producing, at pre-processing, lists of inter-processor boundary cells for each sub-domain.

Depending on the flow field simulated, droplets may be removed from the level set field in some sub-domains and not in others. However, it is crucial to synchronise the re-initialisation of the hyperbolic tangent profile on all the processors or the parallel computation will crash. This problem is mitigated by the creation of a list of booleans storing the state of each sub-domain. Whenever, a droplet is removed from φ in given sub-domain, its state is set to 1 — signifying the need for re-initialisation — and an all-to-all transfer of the sub-domain states is performed. The re-initialisation then occurs on each sub-domain as long as there is at least one state set to 1 in the list.