Supporting Information:

Substrate Wettability Influences Internal Jet Formation and Mixing during Droplet Coalescence

Thomas C. Sykes,[*,†] David Harbottle,[‡] Zinedine Khatir,[¶] Harvey M. Thompson,[§] and Mark C. T. Wilson[*,§]

†*EPSRC Centre for Doctoral Training in Fluid Dynamics, University of Leeds, Leeds LS2 9JT, United Kingdom*

‡*School of Chemical and Process Engineering, University of Leeds, Leeds LS2 9JT, United Kingdom*

¶*School of Engineering and the Built Environment, Birmingham City University, Birmingham B4 7XG, United Kingdom*

§*School of Mechanical Engineering, University of Leeds, Leeds LS2 9JT, United Kingdom*

E-mail: t.c.sykes@outlook.com; m.wilson@leeds.ac.uk

Laplace pressure recovery

The pressure difference across a droplet free surface, Δp due to surface tension is known as the Laplace pressure. For a stationary, free droplet the Laplace pressure is given by

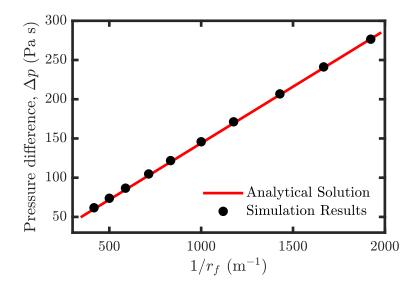$$\Delta p = \frac{2\sigma}{r_f}, \tag{1}$$

Figure S1: Analytical droplet pressure difference across the free surface compared to simulated results for a free droplet of radius r_f after 5×10^{-3} capillary time units, τ_σ.

where σ is the surface tension and r_f is the free droplet radius. As noted in the main text, the gauge pressure (modified pressure in OpenFOAM terminology) was initialised as zero everywhere in this work due to the unknown pressure distribution within the meniscus bridge. Hence, it is important to check that the analytical value of the pressure is correctly recovered in a sufficiently short time so as not to have an undue influence on the dynamics.

In order to validate the recovery of Laplace pressure, a single free spherical water droplet was initialised in the standard numerical setup used for the coalescence results reported. Multiple droplet radii, from 0.5×10^{-3} m to 2.4×10^{-3} m, were considered. Recall that $g_i = 0$ so such a droplet is temporally stable. However, to be consistent across the droplet sizes considered, the Laplace pressure was measured after 5×10^{-3} simulated units on the applicable capillary time scale,

$$\tau_\sigma = \sqrt{\frac{\rho_d r_f^3}{\sigma}}, \tag{2}$$

where ρ_d is the droplet density. For a droplet with $r_f = 1$ mm, $t = 0.005\tau_\sigma \approx 19$ μs, before which time meniscus bridge growth is expected to be negligible. The analytical and simulated mean Laplace pressure within the droplet (defined as cells where $\alpha > 0.9999$) are plotted against the reciprocal of droplet radius in Figure S1. Excellent agreement between

Table S1: Mesh details for the mesh sensitivity analysis. 'Refined cells' refers to cells within refined region of the domain (the entire droplet volume and in the region of its free surface) whilst 'base cells' specifies the minimum resolution in the air. Droplet resolutions are computed with $r_f = 1.14\,\text{mm}$, consistent with the (numerical and laboratory) experiments.

No. refined cells per r_f	No. base cells per r_f	Refinement Levels	Total No. cells at $t = 0.2\,\text{ms}$	Droplet Res. (µm)
64	4	4	0.58×10^6	17.81
64	8	3	0.58×10^6	17.81
80	5	4	1.06×10^6	14.25
80	10	3	1.06×10^6	14.25
96	6	4	1.76×10^6	11.88
96	12	3	1.78×10^6	11.88

the simulated and analytical results is seen, confirming the correct, rapid recovery of Laplace pressure and validating both the surface tension formulation within the numerical framework and the initial condition for pressure.

Mesh sensitivity analysis

To determine the sensitivity to mesh resolution of the results presented in the main text, six meshes with three different numbers of cells per free droplet radius, r_f within the droplet were considered as detailed in Table S1. Two different levels of refinement were considered. For example, a mesh with 10 cells per r_f and 3 levels of refinement (as generally used in this work) have $10 \times 2^3 = 80$ cells per r_f within the whole droplet volume. With each of these six meshes, a similar study to that shown in Figure 3a of the main text was conducted, with $V_s = 3.6\,\text{mm}$, $V_f = 6.2\,\text{mm}$, $\theta_a = 100°$, $\theta_0 = 82°$ and $\theta_r = 75°$. The evolution of spread length, total droplet height and internal interface height are presented in Figures S2, S3 and S4, respectively.

All of the meshes considered are relatively fine within the droplet region (especially compared mesh resolutions generally used in the literature) so it is not surprising that there is no large quantitative difference between any of them in any of the three metrics considered. Even in regions of rapid change where differences are likely to be accentuated as only the
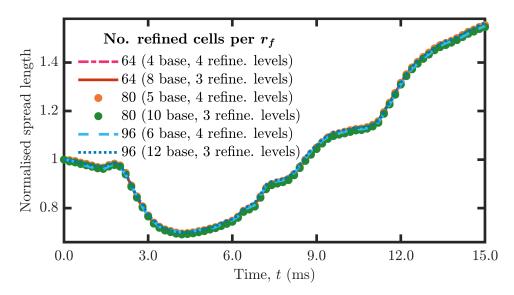
Figure S2: Spread length evolution, normalised by the initial spread length, for six different meshes (detailed in Table S1). $V_s = 3.6\,\mathrm{mm}$, $V_f = 6.2\,\mathrm{mm}$, $\theta_a = 100°$, $\theta_0 = 82°$ and $\theta_r = 75°$. The green filled marker represents the mesh resolution used in this work.

cell-centre values are used to derive the plotted data across a diffuse interface (the interface is defined as either $\alpha = 0.5$ or $\beta = 0.5$, as appropriate), the differences due to the number of cells in the droplet region are small and, in particular, negligible for the conclusions of this study. Moreover, the dynamics were found to be remarkably insensitive to the mesh resolution within the air, characterised by the number of base cells per r_f. Higher numbers of base cells with a constant refined region resolution (achieved by fixing the number of cells within the droplet region by reducing the number of refinement levels) have been considered than are shown here with almost no quantitative difference in any of the three metrics.

In this work, 80 refined cells per r_f were deployed, with a base mesh resolution of 10 cells per r_f, indicated by the green markers in Figures S2, S3 and S4. The mesh sensitivity analysis detailed here confirms that all free surface and internal features are sufficiently resolved with such mesh resolution, whilst further refinement would have had a negligible effect on the results presented in the main text.
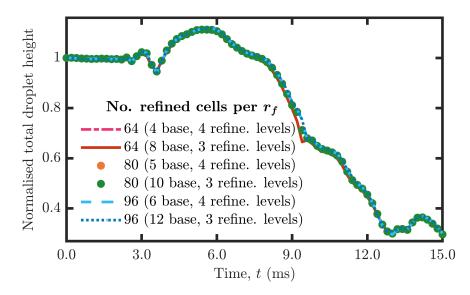
Figure S3: Total droplet height evolution, normalised by the initial total droplet height, for six different meshes (detailed in Table S1). $V_s = 3.6\,\text{mm}$, $V_f = 6.2\,\text{mm}$, $\theta_\text{a} = 100°$, $\theta_0 = 82°$ and $\theta_\text{r} = 75°$. The green filled marker represents the mesh resolution used in this work.

Interface compression velocity

The definition and computation of the compression velocity, $u_{c,i}$ found in the advection-diffusion equation for α (equivalently β) is explained in detail here. In particular, it will be useful for those wishing to implement a conserved scalar within an OpenFOAM solver and therefore should aid in reproducing this work. The conserved scalar is assumed to be the volume fraction, α throughout here, but the passive scalar, β is treated in the same way.

As insinuated in the main text, the compression term has a user-modifiable coefficient, c_α which is taken to be unity in this work. Many such values can be found in the literature, where one is usually the lowest. An optimum choice does not necessarily exist, and the appropriate value can depend on many factors including the underlying flow physics and other solver settings chosen (see below). The value of one was chosen here to avoid over compression that might have an undue influence the delicate free surface dynamics.

In OpenFOAM 4.1, the compression velocity is given by

$$u_{c,i} = c_\alpha \left| \frac{\phi}{\left(S_{f,j} S_{f,j}\right)^{\frac{1}{2}}} \right| \hat{n}_i, \tag{3}$$
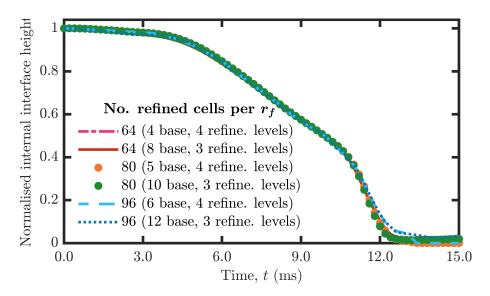
Figure S4: Internal interface height evolution, normalised by the initial internal interface height, for six different meshes (detailed in Table S1). $V_s = 3.6\,\text{mm}$, $V_f = 6.2\,\text{mm}$, $\theta_\text{a} = 100°$, $\theta_0 = 82°$ and $\theta_\text{r} = 75°$. The green filled marker represents the mesh resolution in this work.

where $S_{f,j}$ is the cell face area vector and ϕ is the volumetric flux through the cell face. Within the numerical framework, the velocities in eq 3 are those projected onto the cell faces in the finite volume discretisation. Hence, eq 3 is actually an equation for the face-flux compression velocity which is calculated in the `alphaEqn.H` file of the interFoam solver.

The unit normal to the interface, $\hat{n}_i$ is mathematically defined as

$$\hat{n}_i = \frac{\partial \alpha}{\partial x_i} \bigg/ \left(\frac{\partial \alpha}{\partial x_k} \frac{\partial \alpha}{\partial x_k} \right)^{\frac{1}{2}}. \tag{4}$$

As was the case for velocity though, the cell face values are required in eq 3. Hence, the volume fraction gradients are actually interpolated from the cell centres to the cell faces in the numerical framework before being substituted into eq 4. Note that away from the free surface, the volume fraction gradients should vanish. To stabilise the computation, an additional small term, δ_N is added to the denominator of eq 4, which is given by

$$\delta_N = \varepsilon \left(\frac{\sum_{k=1}^{N} V_{\text{cell},k}}{N} \right)^{-\frac{1}{3}},$$

Table S2: Solver settings for α. Equivalent settings are used for β.

Option	Value	Notes
`nAlphaCorr`	1	
`nAlphaSubCycles`	1	Time step tightly controlled instead of subcycling.
`cAlpha`	1	Small to avoid too much interface compression.
`MULESCorr`	no	Keep MULES explicit rather than semi-implicit.
`nLimiterIter`	1	One MULES iteration over the limiter.
Preconditioner	DIC	
Solver	PCG	
Tolerance	1×10^{-9}	Set low to ensure convergence.
Rel. Tolerance	0	Forces solution to converge to the tolerance above.
Min. Iterations	50	Ensure stable converged solution each time step.
Max. Iterations	1000	Set high so as not to be an effective constraint.

where N is the number of cells, ε is a scaling constant (typically taken as 1×10^{-8}) and $V_{\text{cell},k}$ is the volume of the kth cell. The formulation in brackets represents the average cell volume, so it is clear than δ_N is typically a very small number. The same approach is used to compute the normal for the free surface curvature, κ described in the main text and is implemented in the `interfaceProperties.C` file within the OpenFOAM source code.

Solution and algorithm control

This section lists the linear equation solvers, tolerances and settings used for the simulations in this work, and provides justification/explanation of the choices where appropriate.

In general, the residual tolerance was set very low for all variables to ensure convergence at each time step, thus favouring accuracy over speed. The relative tolerance was always set to zero (i.e. its influence was rescinded) to ensure that the prescribed tolerance was reached. Whilst a maximum number of iterations was set, the value was chosen to be very large so that it was not an effective constraint.

Subcycling of advection-diffusion equations is often performed in OpenFOAM simulations in order to relax the strict Courant number limit on the time step. However, in this work it was preferred to maintain a strict Courant number limit (i.e. 0.15) for the con-

Table S3: Solver settings for `p_rgh` (the modified pressure).

Option	Value	Notes
Smoother	Gauss Seidel	
Solver	GAMG	More efficient (far fewer iterations) than, e.g., PCG.
Tolerance	1×10^{-7}	The final iteration tolerance is 1×10^{-8}.
Rel. Tolerance	0	Forces solution to converge to the tolerance above.
Min. Iterations	2	
Max. Iterations	1000	Set high so as not to be an effective constraint.

tact angle model implementation. Therefore, no subcycling was required. Note that it was crucial to implement the scalar (α and β) advection equations using the Multidimensional Universal Limiter with Explicit Solution (MULES) algorithm to ensure the scalars remain bounded between zero and one. MULES is a flux-corrected transport scheme implemented in OpenFOAM. An explicit version of the MULES solver was maintained, as seen in Table S2.

Table S4: Solver settings for `pcorr` (the pressure correction).

Option	Value	Notes
Smoother	Gauss Seidel	
Solver	GAMG	More efficient (far fewer iterations) than, e.g., PCG.
Tolerance	1×10^{-8}	Set low to ensure convergence.
Rel. Tolerance	0	Forces solution to converge to the tolerance above.
Min. Iterations	2	
Max. Iterations	1000	Set high so as not to be an effective constraint.

In OpenFOAM, the hydrostatic pressure contribution is subtracted from the usual pressure to define a modified pressure, $p' = p - \rho g_j x_j$ for numerical convenience. Hence, details for this modified pressure (called `p_rgh`) are given in Table S3. Table S4 give details of the pressure correction required for the dynamic mesh. The velocity is determined from the updated pressure in each time step with the settings seen in Table S5. A momentum predictor step was employed at the beginning of each time step (see Table S6).

A pressure-implicit with splitting of operators (PISO) algorithm was used to solve the coupled momentum and mass conservation equations, with the controls given in Table S6. No under-relaxation was applied (relaxation factors were set to one).

Table S5: Solver settings for `U` (the velocity).

Option	Value	Notes
Preconditioner	DILU	
Solver	PBiCG	Computational cost not enough to warrant GAMG.
Tolerance	1×10^{-10}	Set low to ensure full convergence.
Rel. Tolerance	0	Forces solution to converge to the tolerance above.
Min. Iterations	5	
Max. Iterations	1000	Set high so as not to be an effective constraint.

Discretisation schemes

Linear interpolation was used for centre-to-face interpolation of values to determine cell face data, whilst a corrected surface-normal gradient scheme was used for gradients, and a corrected scheme was used to determine components of the gradient normal to a cell face (`snGradSchemes`).

Table S7 details the discretisation schemes used. The `Euler` scheme in OpenFOAM is first order implicit and bounded. The limitedLinear scheme has a 'V' added to ensure the direction of steepest gradients were taken into account within the calculation. A special version of the vanLeer scheme was used for the advection of the volume fraction, α and passive scalar, β to bound the blending factor between 0 and 1. In particular, the underlying scheme is vanLeer, but becomes upwind if the blending factor goes out of bounds in order to stabilise the solution. The `Gauss interfaceCompression` scheme is specifically designed for the compression term described above, so should always be used with it.

Table S6: PIMPLE controls (reduced to PISO), controlling the algorithm solving the coupled equations for momentum and mass conservation via a pressure equation.

Option	Value	Notes
Momentum Predictor	yes	
No. Outer Correctors	1	Reduces to PIMPLE to PISO.
No. Correctors	3	
No. Non-Orthogonal Correctors	1	Update explicit non-orthogonal correction.
Mesh Dynamic	yes	Correct pressure for moving mesh.

Table S7: OpenFOAM discretisation schemes used in this work.

Term	OpenFOAM Scheme
Time derivatives (i.e. $\partial/\partial t$)	`Euler`
Gradients (i.e. $\partial/\partial x_i$)	`Gauss linear`
Tensor advection (e.g. $\partial(u_i u_j)/\partial x_j$)	`Gauss limitedLinearV 1`
Viscous stress	`Gauss linear`
Scalar advection	`Gauss vanLeer01`
Scalar compression (anti-diffusion)	`Gauss interfaceCompression`
Laplacian	`Gauss linear corrected`

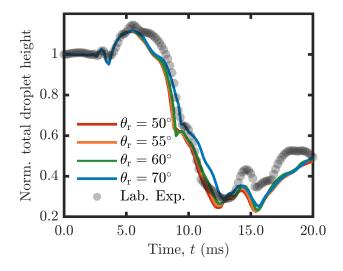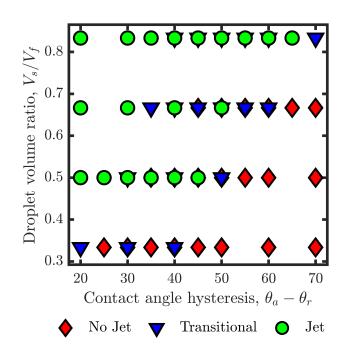# Changing receding contact angle – total droplet height

Figure S5: Total droplet height normalised by initial droplet height at $t = 0\,\text{ms}$. Image-processed experiments with $V_s = 3.9\,\mu\text{L}$ and simulation results (solid lines). $\theta_\text{a} = 100°$ and $\theta_0 = 82°$ in all simulations.

Figure S5 demonstrates the effect of changing the receding contact angle on total droplet height using simulations, accompanying those for the internal interface height and spread length in the main text. The fixed prescribed contact angle values are $\theta_\text{a} = 100°$ and $\theta_0 = 82°$. The droplet volumes are such that no internal jet materialises in the experiment ($V_s = 3.9\,\mu\text{L}$; $V_f = 6.2\,\mu\text{L}$), with the appropriate image-processed data also shown. Excellent agreement between the experiment and the $\theta_\text{r} = 50°$ simulation is evident, with little difference between all of the simulated cases until column collapse (after the maximum height). Indeed, column collapse is slightly delayed for the largest contact angle ($\theta_\text{r} = 70°$) due to the lower minimum

spread length acting to hold the droplet up. Due to the higher contact angle hysteresis, the shoulder (temporary reduction in the rate of height decrease) during column collapse is more prominent for lower receding contact angles. As seen for the $V_s = 5.5\,\mu\text{L}$ case in the main text, increased recoil occurs in the experiment due to the aforementioned differences in spreading, but the general features materialising are very similar. Note that recoil occurs after, and so does not affect, jet formation.

Contact angle hysteresis

Figure S6: Numerically-generated regime map for jet dependence on droplet volume and wettability, plotted against contact angle hysteresis. $\theta_0 = 90°$ and $V_f = 6\,\mu\text{L}$ are fixed. $\theta_a \in \{100°, 110°, 120°\}$ and $\theta_r \in \{50°, 55°, \ldots, 80°\}$, depending on the sessile droplet volume.

The droplet volume ratio against receding contact angle, θ_r regime map in the main text is replotted against contact angle hysteresis, $\theta_a - \theta_r$ in Figure S6. An acute lack of clear regime boundaries is seen, indicating that contact angle hysteresis itself is not an accurate predictor of jet formation. The importance of the advancing contact angle to jet formation is therefore subordinated to that of the receding contact angle.