Spherical harmonics decomposition based on Cartesian level-set field

Ignacio Roa, Jorge César Brändle de Motta, Alexandre Poux

Technical Report

Cite this article: I. Roa Antúnez, J.C. Brändle de Motta, A. Poux (2022). Spherical harmonic decomposition. [Technical Report] CNRS, Normandy Univ., INSA Rouen, CORIA. <hal-04004128>

Subject Areas:
Fluid Mechanics, Computational physics

Keywords:
Two-phase flows, Numerical Simulations, Spherical harmonics, droplet oscillations

Author for correspondence:
Ignacio Roa Antúnez
e-mail: ignacio.roa-antunez@coria.fr
Jorge César Brändle de Motta
e-mail: jorge.brandle@coria.fr
Alexandre Poux
e-mail: alexandre.poux@coria.fr

Spherical harmonics decomposition based on Cartesian level-set field

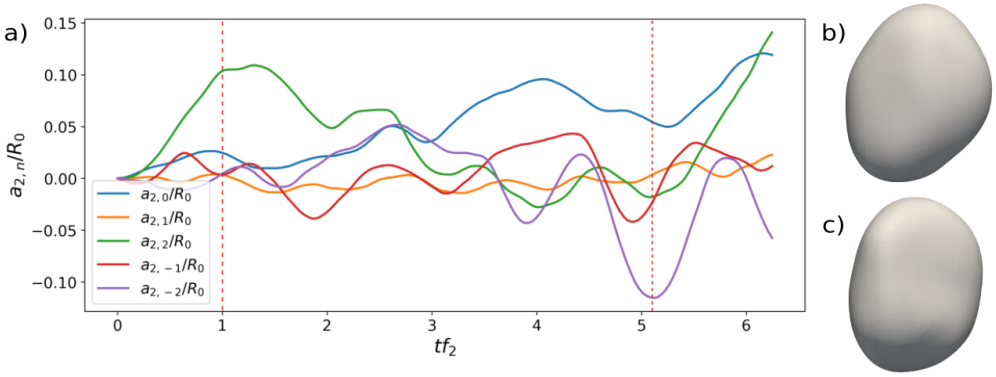
I. Roa Antúnez, J.C. Brändle de Motta, A. Poux

CNRS, Normandy Univ., UNIROUEN, INSA Rouen, CORIA, 76000 Rouen, France

February 2023

Droplet deformation and breakup are important phenomena in the atomization process due to their impact in the final distribution of droplet size. For many applications such as spray painting, coating or fuel injection, it is desirable to be able to obtain appropriate fragmentation sizes. Turbulence is one of the main mechanisms through which secondary breakup occurs. Several authors relate the turbulent droplet breakup to a resonant mechanism reached by their free oscillations.

A theoretical framework for the droplet oscillations for each individual mode using a spherical harmonic decomposition is considered [1], with the help of the library PySthools [2]. This is appraised from direct numerical simulation (DNS) data using the in-house code `archer` and the `spherical_harmonics` library of the post-processing tool `pyarcher`. The present report aims to document the developed numerical tool and its appropriateness on a number of test cases.



a) Temporal evolution of the amplitude coefficients $a_{2,n}$ for degrees $n = -2$ to $n = +2$, where R_0 is the initial radius of the droplet, t is time and f_2 corresponds to the theoretical angular frequency associated to mode 2. b) Snapshot of the droplet at $tf_2 = 1$. c) Snapshot of the droplet at $tf_2 = 5.1$.

Contents

List of Figures

1. Context and objectives

1.1. Spherical harmonics

Any shape that is star shaped (radially convex set) can be described in a spherical coordinate basis. The function $f(\theta, \varphi)$ that describes the shape, evaluated over the surface of a sphere can be expressed as an expansion of spherical harmonics functions known as Laplace series [3]:

$$f(\theta,\varphi) = \sum_{m=0}^{\infty} \sum_{l=-m}^{m} a_{m,l} Y_m^l(\theta,\varphi), \tag{1.1}$$

where $a_{m,l}$ corresponds to the spherical harmonic coefficient, Y_m^l is the spherical harmonic function, θ is the polar angle (co-latitude) and φ is the azimuthal angle (longitude). Finally m and l correspond to the spherical harmonic degree and order respectively, with $m \geq 0$ and $-m \leq l \leq m$. The spherical harmonic degree describes the number of lobes across the interface of the droplet, whilst the order defines its orientation. The spherical harmonic functions can be defined as

$$Y_m^l(\theta,\varphi) = \sqrt{(2m+1)\frac{(m-l)!}{(m+l)!}} P_m^l(cos\theta) e^{il\varphi}, \tag{1.2}$$

with $P_m^l(cos\theta)$ corresponding to an axisymmetric description, also known as Legendre polynomials

$$P_m^l(x) = \frac{(-1)^l}{2^m m!}(1-x^2)^{l/2}\frac{d^{m+l}}{dx^{m+l}}(x^2-1)^m. \tag{1.3}$$

The spherical harmonic functions are orthonormal for all degrees m and orders l:

$$< Y_m^l, Y_m^{l*} >= \int Y_m^l(\theta,\varphi) Y_m^{l*}(\theta,\varphi)\, d\Omega = \delta_{m\,m^*}\delta_{l\,l^*} \tag{1.4}$$

where $d\Omega$ corresponds to the differential of the surface area of the unit sphere $sin\theta\, d\theta\, d\varphi$. Then, by multiplying equation 1.1 by Y_m^{l*} and integrating over the surface, we can show that the spherical harmonic coefficients of a function can be calculated as

$$a_{m,l} = \int_\Omega f(\theta,\varphi) Y_m^l(\theta,\varphi)\, d\Omega. \tag{1.5}$$

Parseval's theorem relates the integral of a function with the sum of the squares if the function's Fourier coefficients. Using the orthogonality properties of the spherical harmonics functions, this relation can be then extended to the spherical geometry. Since we aim to relate the spherical harmonic expansion with the surface deformation, we can express the total deformation as the integral over the surface of the body:

$$\eta_\Omega^2 = \int_\Omega \eta^2 d\Omega \tag{1.6}$$

Which, due to the orthogonality of spherical harmonics (equation 1.4) can be expressed as

$$\eta_\Omega^2 = \sum_{m=0}^{\infty} \sum_{l=-m}^{m} a_{m,l}^2 \tag{1.7}$$

From the link of the spherical harmonic expansion with the total deformation of the surface, we can extrapolate to obtain another parameter defined as the power spectrum. The power spectrum a_m^2 is related to the spherical harmonic coefficients by

$$a_m^2 = \sum_{l=-m}^{m} a_{m,l}^2 \tag{1.8}$$

The expansion will be directly dependent on the reference frame of the system. This means that the results from the expansion will be directly affected by the center of the coordinate system and the reference frame in which we define the angles θ and φ. This in turn will result in the spherical harmonic coefficients varying depending on the before mentioned parameters.

1.2. Objectives of the present work

The objectives of the present technical report are the following:

- Firstly, in §2, a presentation of the `archer` solver and the `pyarcher.spherical_harmonics` library is provided.

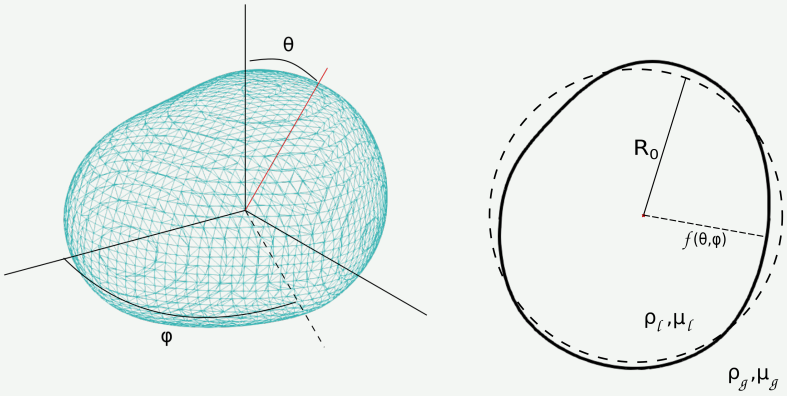


Figure 1: Drop deformation: sketch and definitions.

- A test case providing an example of the use of the implemented library towards the analysis of drops and bubbles deformation is presented in 5.
- Lastly, the order of convergence and the sources of error of the spherical harmonic decomposition with respect to the mesh size are presented in §6.

2. The `archer` solver and the `pyarcher.spherical_harmonics` library

Before proceeding with the validation of the numerical tools used to compute the spherical harmonic decomposition, we start by providing a short description of the `archer` code, the `pyarcher.spherical_harmonics` library and its main functionalities.

2.1. The `archer` solver

The High-Performance-Computing code `archer` developed at the CORIA laboratory is used to obtain the data here used to apply in the spherical harmonic analysis.`Archer` was one of the first codes undertaking the simulation of liquid-jet atomization under a realistic diesel injection configuration [4]. It solves on a staggered cartesian mesh the one-fluid formulation of the incompressible Navier-Stokes equation

$$\partial_t\, \rho \boldsymbol{u} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot (2\mu \mathbf{D}) + \boldsymbol{f} + \gamma \kappa \delta_s \boldsymbol{n} \tag{2.1}$$

where ρ is the density, p the pressure field, μ the dynamic viscosity, $\mathbf{D}$ the strain rate tensor, $\boldsymbol{f}$ a source term, γ the surface tension, $\boldsymbol{n}$ the unit normal vector to the liquid-gas interface, κ its mean curvature and δ_s is the Dirac function characterizing the locations of the liquid gas interface. For solving Eq. (2.1), the convective term is written in conservative form and solved using the improved Rudman's technique [5] presented in Ref. [6]. The latter allows mass and momentum to be transported in a consistent manner thereby enabling flows with large liquid/gas density ratios to be simulated accurately. The viscosity term is computed following the method presented by Ref. [7]. To ensure incompressibility of the velocity field, a Poisson equation is solved. The latter accounts for the surface tension force and is solved using a Multi Grid preconditioned Conjugate Gradient algorithm (MGCG) [8] coupled with a Continuum Surface Force method [9].

For transporting the interface, a coupled level-set and volume-of-fluid (CLSVOF) solver is used, in which the level-set function accurately describes the geometric features of the interface (its normal and curvature) and the volume-of-fluid function ensures mass conservation. The density is calculated from the volume-of-fluid ψ as $\rho = \rho_L \psi + \rho_G (1 - \psi)$, where ρ_L, ρ_G is the density of the liquid and gas phase. The dynamic viscosity (μ_L or μ_G) depends on the sign of the level-set function. In cells containing both a liquid and gas phase, a specific treatment is performed to evaluate the dynamic viscosity, following the procedure of Ref. [7]. For more information about the `archer` solver, the reader can refer to the ARCHER HAL website.

2.2. Reading and pre-processing `archer` files

Results from `archer` simulations are written in `hdf5` format with one time series per processor and multiple time-step per file. For visualization purposes using e.g. `paraview`, `archer` further writes `xmf` files which contain information about the geometry (e.g. origin, grid spacing) and topology (connectivity between `MPI` blocks) of the simulation's domain.

`pyarcher` reads such data using the `dask` library and store them in a `xarray.Dataset`. All the time-steps and all the processors are merged into a 4D array, with coordinates `"x", "y", "z", "t"`. These coordinates can be either specified by the user or read in the `xmf` files. The minimal blocks (one processor and one time-step) are called chunks and are the unit of work for `dask`.

`archer` is a staggered CFD code following the MAC discretization. For some post-computations, the velocity field and scalar field (e.g. the volume-of-fluid, the level-set) should be known at the same point. This is not initially the case due to the staggered nature of the `archer` mesh. In this situation, one needs to interpolate the velocity field at the center of the cells. In `pyarcher`, this is done with the `interp` function of `xarray` which proceeds by linear interpolation. This function has been extended to work with chunked data. It is embedded in the method `.to_cell_center()`.

The data written by `archer` contain ghost cells for boundary conditions and processors interface. The latter are removed during the previously mentioned merging operation, but not the former which can be removed with the method `.without_ghost()`. Boundary conditions (periodic, symmetric) can be prescribed with the method `.add_boundary_conditions()`.

2.3. The `spherical_harmonics` pipeline

The computation of the spherical harmonic coefficients can be done in one line of code through the use of the `spherical_harmonics` method of the `pyarcher.spherical_harmonics` library:

```
decomp = spherical_harmonics.sh_decomp(levelset, vof, center, lmax, ref_radius, reference_frame, recenter, interp_method, normalization, grid_format)
```

Since the decomposition is done in spherical coordinates, the library can only be used for three-dimensional shapes. The function `sh_decomp` takes ten input arguments:

- `levelset` of type `CenteredScalarField` is the scalar field containing the levelset, used to compute the spherical harmonic decomposition.
- `vof` of type `CenteredScalarField`, is the scalar field containing the volume-of-fluid field. This optional argument is only used to compute the center of mass.
- `center` of type `Union[Dict[string, float], string]` can either be a dictionary with the position of the center of the reference frame or a string indicating the desired method to compute the center of the body studied. The methods available are `"cartesian_mean"`, `"mass_center"` and `"surface_center"`.
- `lmax` of type `integer` is the number of desired spherical harmonic degrees to obtain from the decomposition. It defaults to `lmax` $= 6$.
- `ref_radius` of type `float` is the radius of reference of the body. It defaults to `ref_radius` $= 1$.
- `reference_frame` of type `ndarray` corresponds to the reference frame in matrix form. This input determines the orientation of the spherical coordinate system angles $r(\theta, \phi)$. It defaults to `reference_frame` $= ([[1,0,0],[0,1,0],[0,0,1]])$.
- `recenter` of type `bool`. If recenter=True, the levelset will be centered in the domain using as a reference the center computed or given in the `center` argument before computing the decomposition.
- `interp_method` of type `string` corresponds to the interpolation method to use for estimating the radius of the body. Methods `"linear"` and `"cubic"` are supported and taken from the `scipy` library, defaulting to `interp_method` = `"linear"`.
- `normalization` of type `string` corresponds to the normalization method of the spherical harmonic expansion. Supports `"4pi"`, `"schmidt"`, `"unnorm"` and `"ortho"`. By default `normalization` = `"4pi"`.
- `grid_format` of type `string` of type *str* that indicates the format of the sample grid. Supports `"DH"` and `"GLQ"`.

The function `sh_decomp` creates a pipeline constituted of 3 consecutive steps described below:

(i) if the center of the body has not been defined previously and instead a method of computation has been stated as an input, the function `get_center` will be used. This takes as input arguments `center`, `levelset`, `vof` to compute the center of the body using the method available desired. It returns a dictionary with the coordinates of the center in the domain. This center will correspond to the origin of the reference frame. If a dictionary has been defined as input for `center` then this step will be skipped.

(ii) `center_fields`, takes as input argument `levelset` and the output `center` of the `get_center` function described above and positions the center of the body at the center of the domain. In case centering the spheroid does not maintain it inside the domain, it will force the shape to be between its bounds.

(iii) `find_radius_arbitrary_coordinate_system`, takes as input argument the output `levelset` and `center` of the `center_fields` function described above, as well as `theta` $\equiv \theta$, `phi` $\equiv \varphi$, `reference_frame` and `interp_method`. This function returns a value of type `float` corresponding to the radius $r(\theta, \varphi)$ of the

spheroid from the computed center to the interface in the angles θ and φ. This pipeline corresponds to a loop which computes the values of the radius following the spherical coordinates reference frame over an array of angles $r(\theta, \phi)$, which size is given by the `lmax` input from the function `sh_decomp`.

`sh_decomposition` returns a merged `xarray.Dataset` containing the spherical harmonic coefficients arranged by degree and order.

3. Computation of spherical harmonic expansion for a sphere

3.1. Definition

As seen in section 1, the spherical harmonic decomposition corresponds to an expansion of the function that describes the surface over a sphere. With the intent of testing and validating the mentioned framework, a test is performed in a spherical droplet generated using `ARCHER`. This means that for this first validation test, corresponding to a perfect sphere, the result of the expansion should resemble the surface of a unit sphere. This means:

$$\int_{\Omega} f(\theta, \varphi) Y_m^l(\theta, \varphi)\, d\Omega = 1 \tag{3.1}$$

3.2. Description of the validation test case

With the aim of showing the accuracy obtained by the spherical harmonic decomposition, we use the following benchmark:

- A level-set function field representing a sphere of radius $R_s = 1$ is set. The numerical domain contains $N_x \times N_y \times N_z = 64^3$ points and the grid spacing is $\Delta x = \Delta y = \Delta z = 3 \times R_s / N_x$. The sphere is placed at the center of the numerical domain.
- The `spherical_harmonics` library of `pyarcher` is then used to compute the center and radius of the sphere, and the spherical harmonic expansion of the 3D body.
- We show the errors obtained due to the computation of the radius using the three different available methods, as well as plotting the grid of obtained radius from the `find_radius_arbitrary_coordinate_system` function.

3.3. Results

As previously mentioned, to compute the spherical harmonic expansion of a three-dimensional body, we first fill out a grid with the radius of the spheroid at given angles θ, φ. With this information, it will be possible to reconstruct the shape of the body and do the corresponding expansion. In figure 2 it is possible to observe the mapping of the radius of the spherical shape. It is noticeable that there are numerical errors up to an order of $1E-8$, which shows on the mapping of the radius grid. This has consequences when estimating the spherical harmonic coefficients $a_{m,l}$.

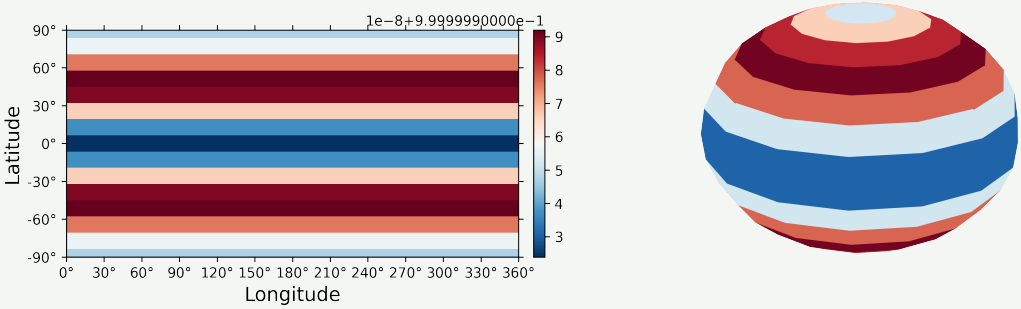


Figure 2: Radius mapping and reconstruction of a spherical droplet with a mesh size of 64^3, using the cubic interpolation method with a DH grid.

Results of spherical harmonic decomposition for a sphere up until degree $m = 6$ are presented in fig 3. As expected, since we are doing the expansion for a sphere, we simply obtain the value of the spherical harmonic coefficient corresponding to a degree and order of $m = 0, l = 0$ respectively. It is also possible to graph the power per degree of spherical harmonic, as a way to note the errors in the decomposition. We may considerate the values accompanying the modes $m \geq 1$ to be numerical error.

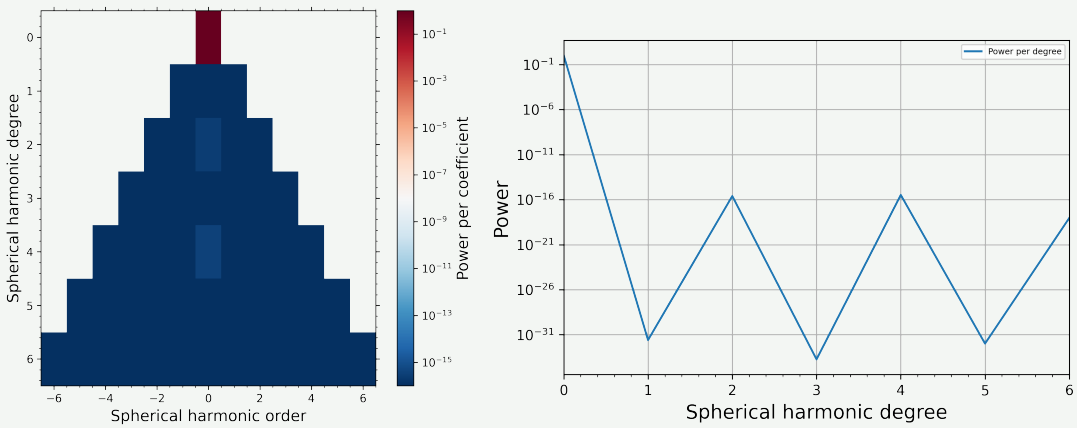


Figure 3: Spherical harmonics spectrum and power spectrum of a sphere for degrees $m = 0$ to $m = 6$.

4. Decomposition of spherical harmonic modes

This second validation example corresponds to the decomposition of a droplet in which we insert a perturbation of small amplitude, corresponding to a spherical harmonic function Y_m^l, with the goal of portraying the accuracy of the decomposition for symmetric and asymmetric cases, using for this the most prominent eigen-modes.

When taking the analysis to a real-case scenario, where a droplet or bubble oscillates due to the effect of its environment, the most significant ones correspond to orders $2 \leq m \leq 4$ being the influences of $m > 4$ negligible, while $m = 1$ is associated with transport, making it desirable to neglect the coefficients $a_{1,l}$ to be able to correlate the surface change with purely its deformation.

4.1. Description of the validation test case

With the aim of showing the accuracy of the spherical harmonic decomposition pipeline, we use the following benchmark:

- A level-set function field representing a sphere of radius $R_s = 1$ plus the insertion of a spherical harmonic function Y_m^l of amplitude $\eta = 0.2$ for $m = 2, 3, 4$ is set. The numerical domain contains $N_x \times N_y \times N_z = 64^3$ points and the grid spacing is $\Delta x = \Delta y = \Delta z = 4.5 \times R_s / N_x$. The body is placed at the center of the numerical domain.
- The `spherical_harmonics` library of `pyarcher` is then used to compute the center and radius of the body, and its spherical harmonic decomposition.
- We show the errors associated with an inaccurate computation of the body's center and the accuracy of the method.

4.2. Results

As portrayed with the spherical case, we display the radius mapping for the studied modes. For this particular validation case the reference center is computed using the `"mass_center"` method, as it is the most accurate for asymmetric geometries (this will be discussed in section 6). We expect the function to properly decompose the droplet into the inserted perturbation as to be able to expand this formulation into an oscillating drop. From the computation we can preliminary observe that the shape of a droplet perturbed with an eigen-mode 3 is very well portrayed in the reconstruction of the grid (figure 4). Once the mapping has been done, the `spherical_harmonics` function will proceed to compute the decomposition.

In figure 5 the results of the expansion in spherical harmonics as well as the error in accuracy are shown. It is observable that, the more complex the geometry is, the more inaccurate the results will be. As seen for the decomposition for the drop initialized with a deformation of amplitude $\eta = 0.2 Y_3^0$, the error can reach values of an order of magnitude close to 10^2.

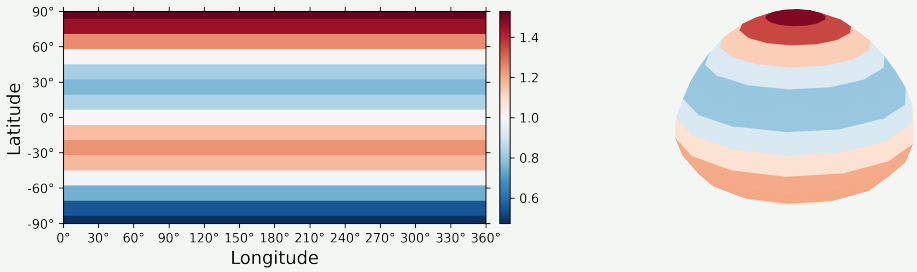


Figure 4: Radius mapping and reconstruction of a sphere of radius 1 plus a Y_3^0 with a mesh size of 64^3, using the cubic interpolation method with a DH grid and its center of mass as origin.

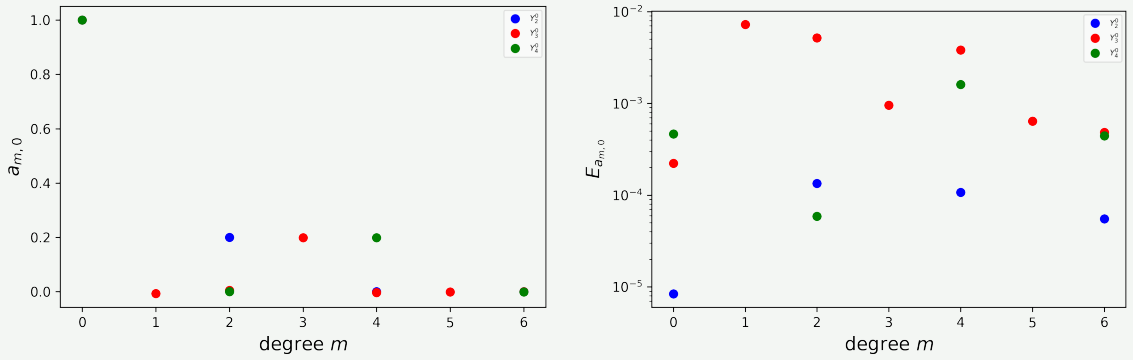


Figure 5: Decomposition in spherical harmonics of validation cases (left) with modes Y_2^0, Y_3^0, Y_4^0 for a grid of 64^3. On the right, the error of said decomposition by degree m.

5. Spherical harmonic decomposition for a droplet in a turbulent flow

5.1. Description of the case

With the aim of showing the results obtained by the spherical harmonic decomposition in a droplet immersed in a turbulent flow, we use the following parameters:

- A level-set function field representing a sphere of radius $R_s = 3.5E-5$ is set. The numerical domain contains $N_x \times N_y \times N_z = 64^3$ points and the grid spacing is $\Delta x = \Delta y = \Delta z = L_x/N_x$. The body is placed at the center of the numerical domain.
- The `spherical_harmonics` library of `pyarcher` is then used to compute the center and radius of the body, and its spherical harmonic decomposition.
- We show the errors associated with an inaccurate computation of the body's center and the accuracy of the method.

5.2. Results

For this particular example case the interest is focused on the droplet oscillations product of the ambient turbulent flow, thus the temporal evolution of the spherical harmonic decomposition is studied for a sphere initially unperturbed. For this, a cubic interpolation method is used, along with the "`mass_center`" method to compute the center at each time step. The temporal evolution of the computation of the power spectrum and the spherical harmonic coefficients $a_{2,l}$ are shown in figure 6. The `pyarcher.spherical_harmonics` library does a good job of computing the spherical harmonic decomposition for each time step of the droplet oscillations up until shortly before the breakup, where the deformation its too high and the convex-set domain constraint is not satisfied anymore.

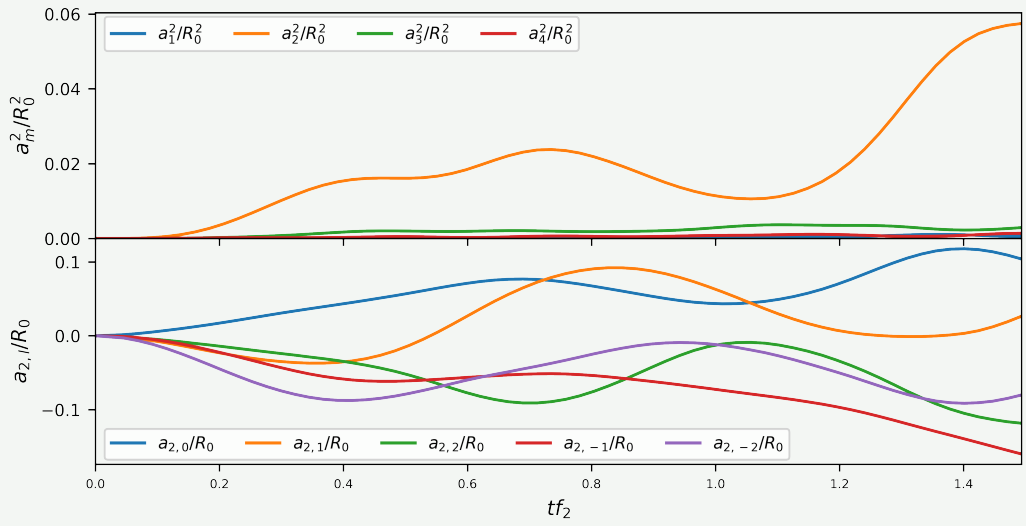


Figure 6: Evolution of the spherical harmonic coefficients of a droplet immersed in a turbulent flow.

6. Order of convergence and sources of error

Firstly, due to the orthogonality properties of the spherical harmonics, the power spectrum of the Laplace expansion should remain constant for any given rotation of the reference frame. As a mean of making sure this mathematical property is maintained by the current framework in figure 7 we present the power spectrum of a time-step of the droplet immersed in the turbulent background flow, showing that the power spectrum is conserved for three different reference frames, in which we rotate the spherical coordinate changing the reference polar and azimuthal angles such that for each reference **i**

$$\theta_{ref_i} = \pi * i/4 + \pi/2$$
$$\phi_{ref_i} = 2\pi * i/2$$

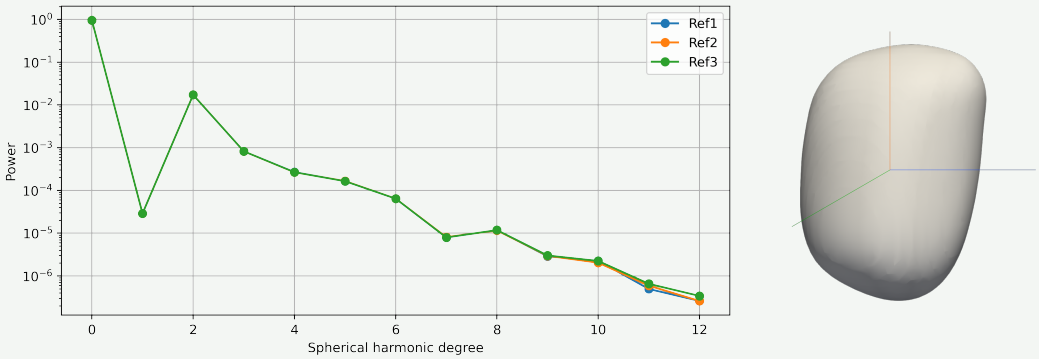


Figure 7: Comparison of the power spectrum of a deformed droplet for three different reference frames

From figure 2 it is possible to notice that the radius is not even throughout the sphere, having a slight numerical error in the computation. This is due to the reconstruction of the levelset and computation of the distance between center and interface at angles θ, φ. For this reason, we perform a simple sensibility analysis to evaluate the error in the calculation of the radius using both available interpolation methods, for each different method for finding the center of the sphere of the validation case presented in section 3. We can observe from the results presented in figure **??** that the

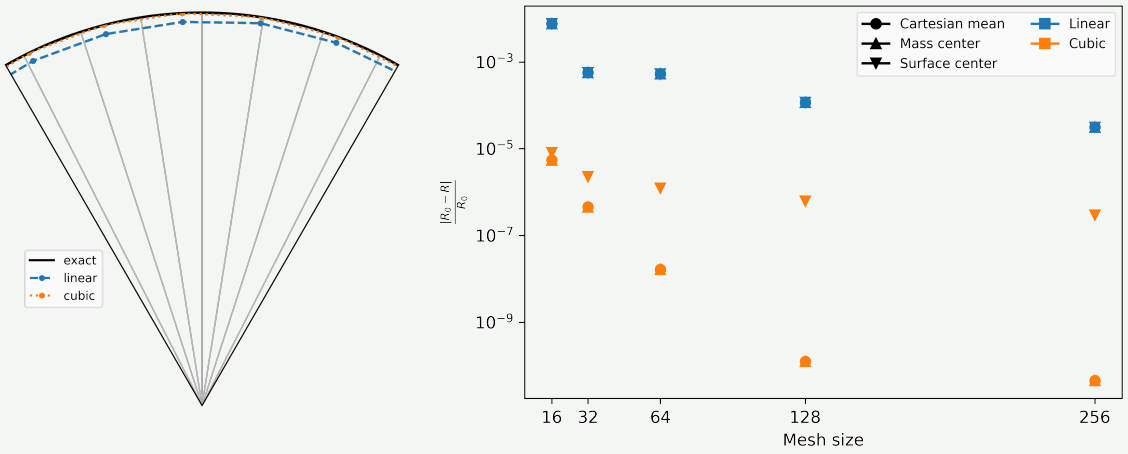


Figure 8: Interface reconstruction comparison between the linear and cubic interpolation method. On the right, we show the error on the calculation of the radius using the same method.

error for the radius using a linear interpolation method is quite big, even for finer mesh sizes. The resolution improves greatly with a cubic interpolation. We can observe that already for grids of size 64^3 the convergence is good.

We must also consider that for a proper computation of the spherical harmonic expansion it is of upmost importance to correctly select the body's center. This aspect is quite challenging because we aim to reduce the influence of the eigen-mode 1, while also maintaining coherence of the center as the droplet deforms. To show the difference in the computation of the coefficients $a_{m,l}$ we perform the spherical harmonic decomposition of the droplets presented in 4 using the different available methods of setting the center. It is expected that for spherical droplets affected by a symmetric deformation (e.g. Y_0^0, Y_2^0, Y_4^0) the computation of the center is quite straight forward, obtaining similar results for the three methods given the symmetry of the body. In contrast, if the perturbation is asymmetrical (e.g. Y_3^0), the `"cartesian_mean"` method will give different result.

We can observe in figure 9 the difference in the decomposition for a droplet shaped as a mode 3 (Y_3^0). It is noticeable that doing the decomposition without an appropriate center can greatly alter the result. Nonetheless, it is important to note that this does not mean that the calculation is wrong. Since the Laplace expansion over the surface of the sphere depends on its reference frame, where the center is placed will affect directly the way the `find_radius_arbitrary_coordinate_system` generates the grid.

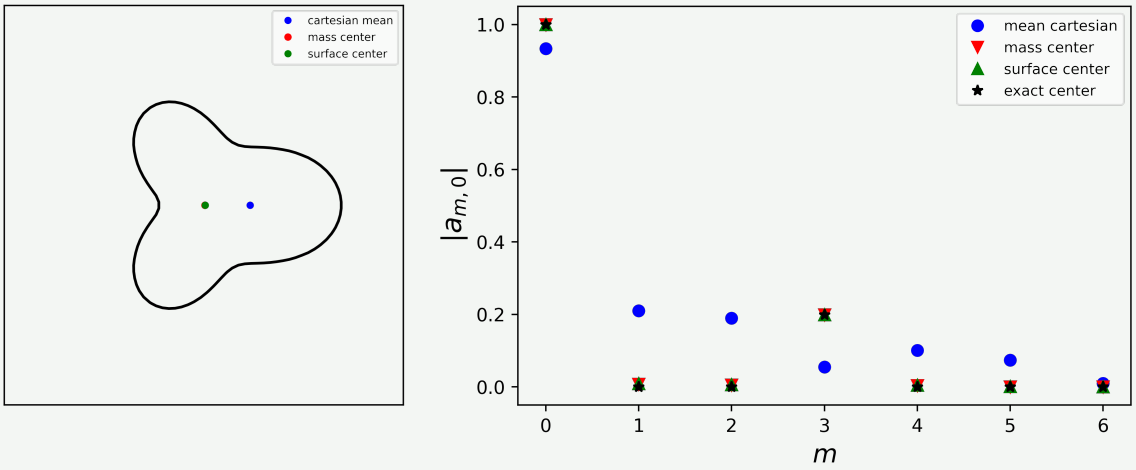


Figure 9: Computation of the center with different methods for a droplet with a deformation Y_3^0 of amplitude $\eta = 0.2$. On the right, a plot demonstrating the difference in the decomposition using the available methods for computing the center.

As mentioned earlier, this framework is limited to convex-set domains. This in consequence does not allow the user to characterize the high deformation encountered shortly before the breakup

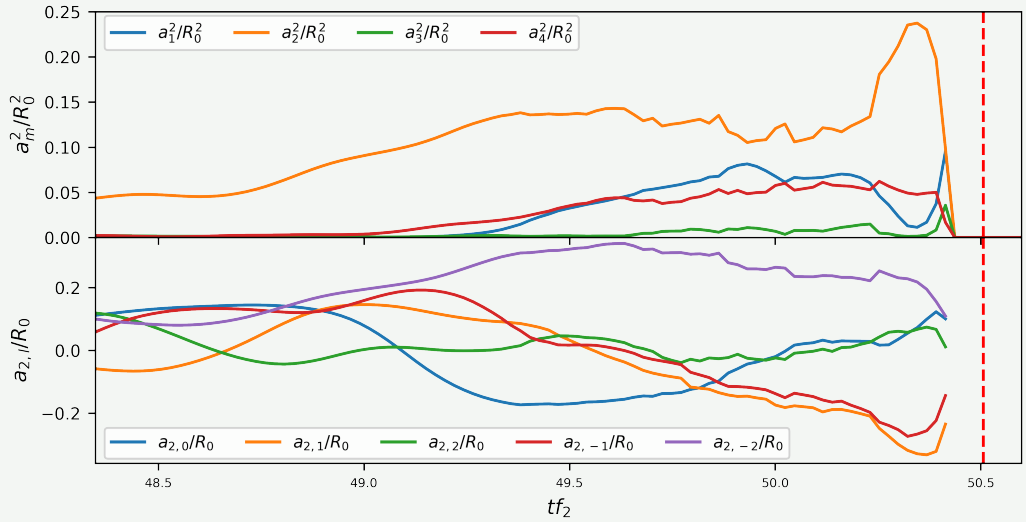


Figure 10: Limitation due to non-convex-set domain. Here, the red dotted line represents the time were the droplet breaks

7. Conclusion

This report provides an in-depth description of the `pyarcher.spherical_harmonics` library, aiming at documenting the built-in functionalities as well as providing some validation test cases allowing to show the accuracy of the numerical procedures for computing the center of a three-dimensional body and its spherical harmonic decomposition, the impact of spatial resolution and the constraints and points to consider when working with it. An example case of applications for the study of droplet oscillations is also provided, showing that this kind of analysis can be useful for further digging into atomization related topics. To summarize the present analysis, three different points are presented:

SUMMARY

(1) This procedure for computing the spherical harmonic decomposition is developed thanks to the PySthools library [2] and proves to be an efficient method for computing the spherical harmonic expansion of any three-dimensional star-convex set body.

(2) Computing the spherical harmonic decomposition with an accurate center of the body is critical when performing the numerical calculation. Failing to do so will lead to inaccurate results, as seen in figure 9. It is important to use the appropriate center, or in the other hand, select the most effective method to compute it depending on the estimated shape of the body. In any case, it is advised to work with `"mass_center"` as the default option.

(3) This method for computing the spherical harmonic decomposition composes an useful tool for studying the evolution of a three-dimensional body deformation, as seen in the example shown in section 5. This gives a quick and accurate description of the shape and the modes of spherical harmonics present, allowing us to identify the oscillations of the different modes of deformation.

References

1. Perrard, S, Rivière, A, Mostert, W, Deike, L. 2021 Bubble deformation by a turbulent flow. *Journal of Fluid Mechanics*, **920**.
2. Wieczorek, M. A., Meschede, M. 2018 SHTools: Tools for working with spherical harmonics. Geochemistry, Geophysics, Geosystems **19.8**, 2574–2592.
3. Arfken, G. B., & Weber, H. J. (1999). Mathematical methods for physicists.
4. Ménard T, Tanguy S, Berlemont A. 2007 Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet. *Int. J. Multiphase Flow* **33**, 510–524.
5. Rudman M. 1998 A volume-tracking method for incompressible multifluid flows with large density variations. *Int J Numer Methods Fluids* **28**, 357–378.
6. Vaudor G, Ménard T, Aniszewski W, Doring M, Berlemont A. 2017 A consistent mass and momentum flux computation method for two phase flows. Application to atomization process. *Computers Fluids* **152**, 204–216.
7. Sussman M, Smith K, Hussaini M, Ohta M, Zhi-Wei R. 2007 A sharp interface method for incompressible two-phase flows. *J Comput Phys* **221**, 469–505.
8. Zhang J. 1996 Acceleration of five-point red-black Gauss-Seidel in multigrid for Poisson equation. *Appl. Math. Comput.* **80**, 73–93.
9. Brackbill, J U, Kothe, D B, Zemach, C. 1992 A continuum method for modeling surface tension. *J Comput Phys* **100**, 335–354.