WPMST1-PR(18) 21377

E. Zoni et al.

Solving hyperbolic-elliptic problems on singular mapped disk-like domains with the method of characteristics and spline finite elements

Preprint of Paper to be submitted for publication in
Journal of Computational Physics

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Solving hyperbolic-elliptic problems on singular mapped disk-like domains with the method of characteristics and spline finite elements

Edoardo Zoni[a,b,*], Yaman Güçlü[a]

[a]*Max-Planck-Institut für Plasmaphysik, Boltzmannstraße 2, Garching 85748, Germany*
[b]*Technische Universität München, Zentrum Mathematik, Boltzmannstraße 3, Garching 85748, Germany*

ARTICLE INFO

Keywords: Vlasov, Poisson, semi-Lagrangian, smooth, spline, finite element

ABSTRACT

A common strategy in the numerical solution of partial differential equations is to define a uniform discretization of a tensor-product multi-dimensional logical domain, which is mapped onto a physical domain through a given coordinate transformation. By extending this concept to a multi-patch setting, simple and efficient numerical algorithms can be employed on relatively complex geometries. The main drawback of such an approach is the inherent difficulty in dealing with singularities of the coordinate transformation.

This work suggests a comprehensive strategy for dealing with the most common situation of disk-like domains with a singularity at a unique pole, where one edge of the rectangular logical domain is collapsed into one point of the physical domain (e.g. a circle). We present robust numerical algorithms for the solution of Vlasov-like hyperbolic equations coupled to Poisson-like elliptic equations in such geometries. We describe a semi-Lagrangian advection solver that employs a new set of coordinates to integrate the characteristic equations in the whole domain, including the pole, and a finite element Poisson solver based on C^1 smooth polar splines (Toshniwal et al., 2017 [1]). The 2D guiding-center model for magnetized plasmas, mathematically equivalent to a vorticity model for incompressible inviscid Euler fluids, is then considered to set up numerical test cases coupling the two solvers. The numerical methods presented show high-order convergence in the space discretization parameters, uniformly across the whole domain, with no order reduction effects due to the singularity. Dedicated examples show that the techniques described can be applied straightforwardly also in the presence of point charges (or, similarly, point-like vortices) within the context of particle-in-cell methods.

*Corresponding author: edoardo.zoni@ipp.mpg.de

November 19, 2018

1. Introduction

This work is concerned with the solution of coupled hyperbolic and elliptic partial differential equations on disk-like domains with a singularity at a unique pole, where one edge of the rectangular logical domain $\Lambda = [0,1]\times[0,2\pi)$ is collapsed into one point of the physical Cartesian domain Ω via a given coordinate mapping $\boldsymbol{F} : \Lambda \mapsto \Omega$ [1]. Our 2D working model is the guiding-center model [2, 3]

$$\begin{cases} \dfrac{\partial \rho}{\partial t} - E^y \dfrac{\partial \rho}{\partial x} + E^x \dfrac{\partial \rho}{\partial y} = 0 \\ -\Delta \phi = \rho \end{cases} , \tag{1}$$

with homogeneous Dirichlet boundary conditions $\phi(\partial\Omega) = 0$. This is tipically used to describe low-density non-neutral plasmas [4, 5, 6, 7] in a uniform magnetic field $\boldsymbol{B} = \hat{z}$, where $\hat{z}$ is a unit vector perpendicular to the (x, y) plane, $\rho = \rho(t, x, y)$ is the density distribution function of the plasma charges, $\phi = \phi(t, x, y)$ is the electric scalar potential associated to the electric field $\boldsymbol{E} = (E^x, E^y) = -\boldsymbol{\nabla}\phi = (-\partial\phi/\partial x, -\partial\phi/\partial y)$ created by the plasma charges and Δ represents the Laplace operator. The 2D velocity field $(-E^y, E^x)$, responsible for the transport of the density distribution function ρ in (1), represents the $\boldsymbol{E}\times\boldsymbol{B}$ drift velocity. System (1) is also mathematically equivalent to the 2D Euler equations for incompressible inviscid fluids, with $-\rho$ representing the vorticity of the fluid and ϕ a stream function. Therefore, (1) has been also used in the fluid dynamics community for a variety of studies related to vortex dynamics and turbulence [8, 9, 10, 11]. The simplest domain we will consider is a circle, which corresponds typically to a 2D cut of some cylindrical 3D physical setup. Examples of more complex domains will be discussed in section 2 and may arise naturally in the context of plasma models as 2D cuts of complex-shaped toroidal devices (e.g. D-shaped Tokamaks [12]).

Regarding the numerical solution of (1), we are interested in solving the hyperbolic part (transport advection equation for ρ) with the method of characteristics and the elliptic part (Poisson's equation for ϕ) with a finite element method based on B-splines. More precisely, the advection equation is solved by computing ρ on a grid following the characteristics backward in time for a single time step and interpolating at the foot of the characteristics using the grid values of ρ at the previous time step. This method is referred to as backward semi-Lagrangian method and was originally investigated in the context of the vorticity advection equation for the barotropic forecasting model by [13] and later for Vlasov-like transport equations by [14, 15, 16, 17, 18, 19]. One advantage of the semi-Lagrangian method is avoiding any limitation related to the Courant-Friedrichs-Lewy (CFL) condition [20] in the region close to the pole, where the grid cells become smaller and smaller. As already pointed out, this scheme works on a structured logical mesh, which is usually constructed to be conformal to the level curves of some given function (in the physical applications, they may correspond to magnetic field flux surfaces for plasma models or level curves of the stream function for fluid models). Since the method is based on the integration of the characteristics backward in time, the choice of coordinates to be used while performing this integration turns out to be crucial: such coordinates need to be well-defined in the whole domain, including the pole. The choice we propose, described in details in section 4, meets this requirement without affecting the robustness, efficiency and accuracy of the numerical scheme. The same

coordinates can be used as well for the forward time integration of the characteristic trajectories of point charges or point-like vortices.

On the other hand, the elliptic Poisson equation is solved with a finite element method based on B-splines. Again, to avoid problems in the region close to the pole, we require the advection field $(-E^y, E^x)$ to be at least continuous there. This means that ϕ, from which E^x and E^y are obtained by means of derivatives, has to be at least of class C^1 in the neighborhood of the pole (no such issues occur anywhere else far from the pole). For this purpose, we follow the approach recently developed by [1] to define a set of globally C^1 smooth spline basis functions on singular disk-like domains. A higher degree of smoothness, consistent with the spline degree, may be imposed as well, if needed.

This paper is organized as follows. Sections 2 and 3 define the types of singular mapped disk-like domains we are interested in and their discrete representation. Sections 4 and 5 describe our hyperbolic advection solver and elliptic Poisson solver, including some numerical tests. Section 6 describes the coupling of the two solvers and discusses a variety of test cases in both circular and more complex geometries. A short appendix describes in details the implementation of the finite element Poisson solver.

2. Singular mapped disk-like domains

We consider polar domains defined by mappings with a singularity at a unique pole. We denote by $(s, \theta) \in [0, 1] \times [0, 2\pi)$ the logical coordinates on the polar domain and by $(x, y) \in \Omega$ the physical Cartesian coordinates. In the following, two analytical examples of such mappings are provided. The first mapping is defined by [21]

$$\begin{aligned} x(s, \theta) &= \bar{x} + (1 - \kappa) s \cos\theta - \Delta\, s^2\,, \\ y(s, \theta) &= \bar{y} + (1 + \kappa) s \sin\theta\,, \end{aligned} \tag{2}$$

where κ and Δ represent the elongation and the Shafranov shift, respectively. The mapping collapses to the pole $(\bar{x}, \bar{y})$ for $s = 0$: $(x(0, \theta), y(0, \theta)) = (\bar{x}, \bar{y})$ for all θ. The Jacobian matrix of the mapping reads

$$J = \begin{bmatrix} (1 - \kappa)\cos\theta - 2\,\Delta\, s & (\kappa - 1) s \sin\theta \\ (1 + \kappa)\sin\theta & (1 + \kappa) s \cos\theta \end{bmatrix},$$

with determinant

$$\det J = s(1 + \kappa)[(1 - \kappa) - 2\,\Delta\, s \cos\theta]\,.$$

The Jacobian determinant vanishes at the pole. Therefore, the Jacobian matrix of the inverse transformation becomes singular there:

$$J^{-1} = \frac{1}{s(1 + \kappa)[(1 - \kappa) - 2\,\Delta\, s\cos\theta]} \begin{bmatrix} (1 + \kappa) s \cos\theta & (1 - \kappa) s \sin\theta \\ -(1 + \kappa)\sin\theta & (1 - \kappa)\cos\theta - 2\,\Delta\, s \end{bmatrix}.$$

The second mapping is defined by [22]

$$\begin{aligned} x(s, \theta) &= \frac{1}{\varepsilon}\left(1 - \sqrt{1 + \varepsilon(\varepsilon + 2\, s\cos\theta)}\right), \\ y(s, \theta) &= \bar{y} + \frac{e\,\xi\, s \sin\theta}{2 - \sqrt{1 + \varepsilon(\varepsilon + 2\, s\cos\theta)}} = \bar{y} + \frac{e\,\xi\, s\sin\theta}{1 + \varepsilon\, x(s, \theta)}\,, \end{aligned} \tag{3}$$

where ε and e represent the inverse aspect ratio and the ellipticity, respectively, and ξ is defined as $\xi = 1/\sqrt{1-\varepsilon^2/4}$. The mapping collapses to the pole $((1-\sqrt{1+\varepsilon^2})/\varepsilon, \bar{y})$ for $s = 0$: $(x(0,\theta), y(0,\theta)) = ((1-\sqrt{1+\varepsilon^2})/\varepsilon, \bar{y})$ for all θ. The Jacobian matrix of the mapping reads

$$J = \frac{e\,\xi}{1+\varepsilon\,x(s,\theta)} \begin{bmatrix} -\dfrac{1+\varepsilon\,x(s,\theta)}{1-\varepsilon\,x(s,\theta)}\dfrac{\cos\theta}{e\,\xi} & \dfrac{1+\varepsilon\,x(s,\theta)}{1-\varepsilon\,x(s,\theta)}\dfrac{s\sin\theta}{e\,\xi} \\ \sin\theta + \dfrac{\varepsilon\,s\sin\theta\cos\theta}{1-\varepsilon^2\,x^2(s,\theta)} & s\cos\theta - \dfrac{\varepsilon\,s^2\sin^2\theta}{1-\varepsilon^2\,x^2(s,\theta)} \end{bmatrix}$$

with determinant

$$\det J = \frac{e\,\xi\,s}{\varepsilon^2\,x^2(s,\theta)-1}\,.$$

Again, the Jacobian determinant vanishes at the pole. Therefore, the Jacobian matrix of the inverse transformation becomes singular there:

$$J^{-1} = \frac{\varepsilon^2\,x^2(s,\theta)-1}{e\,\xi\,s} \begin{bmatrix} s\cos\theta - \dfrac{\varepsilon\,s^2\sin^2\theta}{1-\varepsilon^2\,x^2(s,\theta)} & -\dfrac{1+\varepsilon\,x(s,\theta)}{1-\varepsilon\,x(s,\theta)}\dfrac{s\sin\theta}{e\,\xi} \\ -\sin\theta - \dfrac{\varepsilon\,s\sin\theta\cos\theta}{1-\varepsilon^2\,x^2(s,\theta)} & -\dfrac{1+\varepsilon\,x(s,\theta)}{1-\varepsilon\,x(s,\theta)}\dfrac{\cos\theta}{e\,\xi} \end{bmatrix}$$

In all examples considered in this work the mapping (2) is set up with the parameters shown in Table 1 and the

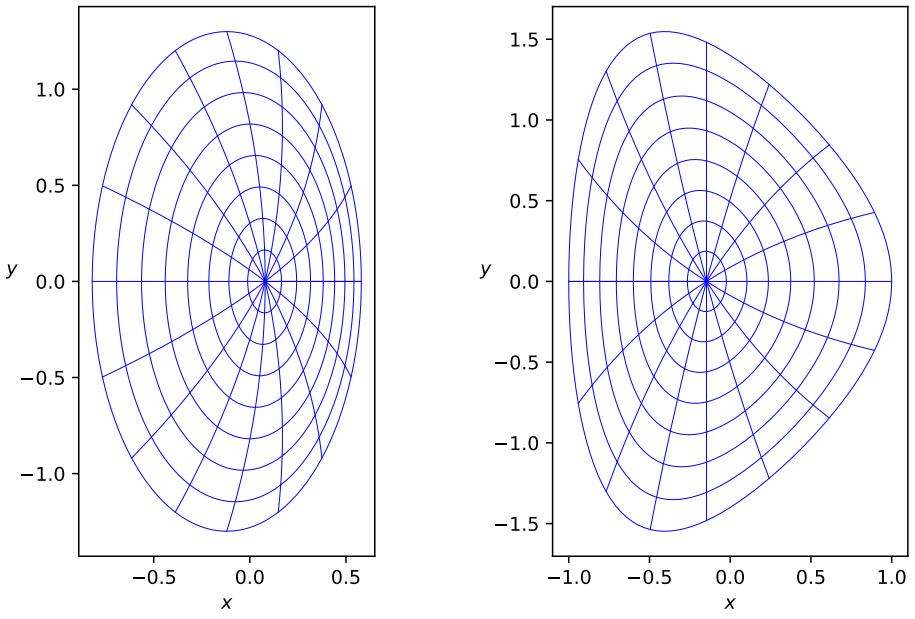

Fig. 1: Singular disk-like domains: mapping (2) (left) and (3) (right). Lines originating from the pole are isolines at constant θ, lines concentric around the pole are isolines at constant s.

mapping (3) is set up with the parameters shown in Table 2. Figure 1 shows the physical domains obtained with these mappings.

$\bar{x}$	$\bar{y}$	κ	Δ
0.08	0	0.3	0.2

Table 1: Parameters used to initialize the mapping (2).

$\bar{y}$	ε	e
0	0.3	1.4

Table 2: Parameters used to initialize the mapping (3).

3. Discrete spline mappings

In practical applications it may not be possible to have an analytical description of the mapping that represents the physical domain of interest, as for the examples discussed above. Moreover, as already pointed out in the introduction, we are going to solve the elliptic equation in (1) with a finite element method based on a set of globally C^1 smooth spline basis functions, obtained from standard tensor-product splines by imposing appropriate constraints [1]. Our numerical method is therefore based on a machinery inherently defined at the discrete level. Hence, we need to have a discrete counterpart of the analytical singular mapped disk-like domains discussed in the previous section.

We start by defining a 2D tensor-product spline basis $\{B_{i_1 i_2}(s,\theta) = B^s_{i_1}(s) B^\theta_{i_2}(\theta)\}_{i_1,i_2}$, with $i_1 = 1,\dots,n_1$ and $i_2 = 1,\dots,n_2$, of B-splines of degree p_1 with open knots in $s \in [0,1]$ and 2π-periodic B-splines of degree p_2 in $\theta \in [0, 2\pi)$. The domain along each direction is decomposed into 1D intervals, also referred to as cells, whose limit points are known as break points. More precisely, the domain along s is decomposed into $n^c_1 = n_1 - p_1$ cells with $n^b_1 = n^c_1 + 1 = n_1 - p_1 + 1$ break points $s_1,\dots,s_{n^b_1}$ and the domain along θ is decomposed into $n^c_2 = n_2$ cells with $n^b_2 = n^c_2 + 1 = n_2 + 1$ break points $\theta_1,\dots,\theta_{n^b_2}$. From the break points, we define a knot sequence of $n^b_1 + 2p_1$ open knots along s,

$$\underbrace{s_1\,,\,\dots\,,\,s_1}_{p_1}\,,\; s_1\,,\,\dots\,,\,s_{n^b_1}\,,\;\underbrace{s_{n^b_1}\,,\,\dots\,,\,s_{n^b_1}}_{p_1}\,,$$

and a knot sequence of $n^b_2 + 2p_2$ periodic knots along θ,

$$\underbrace{\theta_{n^b_2-p_2+1}\,,\,\dots\,,\,\theta_{n^b_2}}_{p_2}\,,\;\theta_1\,,\,\dots\,,\,\theta_{n^b_2}\,,\;\underbrace{\theta_1\,,\,\dots\,,\,\theta_{p_2}}_{p_2}\,.$$

Due to the open knot sequence, the basis functions $B^s_{i_1}(s)$ satisfy the following properties:

$$\begin{aligned} &B^s_1(0) = 1\,, \quad B^s_{i_1}(0) = 0 \quad \text{for} \quad 2 \le i_1 \le n_1\,, \\ &B^s_{n_1}(1) = 1\,, \quad B^s_{i_1}(1) = 0 \quad \text{for} \quad 1 \le i_1 \le n_1 - 1\,. \end{aligned} \tag{4}$$

Moreover, their derivatives satisfy the following property:

$$(B^s_1)'(0) = -(B^s_2)'(0) \neq 0\,, \quad (B^s_{i_1})'(0) = 0 \quad \text{for} \quad 3 \le i_1 \le n_1\,.$$

We now define a discrete counterpart of our analytical singular mapped disk-like domains based on this spline basis. We denote such discrete mappings by $\boldsymbol{F} : [0,1] \times [0, 2\pi) \mapsto \Omega$ and define them as $\boldsymbol{F}(s,\theta) = (x(s,\theta), y(s,\theta))$, with

$$x(s,\theta) = \bar{x}\, B_1^s(s) + \sum_{i_1=2}^{n_1} \sum_{i_2=1}^{n_2} c^x_{i_1 i_2} B^s_{i_1}(s) B^\theta_{i_2}(\theta)\,,$$

$$y(s,\theta) = \bar{y}\, B_1^s(s) + \sum_{i_1=2}^{n_1} \sum_{i_2=1}^{n_2} c^y_{i_1 i_2} B^s_{i_1}(s) B^\theta_{i_2}(\theta)\,,$$

where $(\bar{x}, \bar{y})$ represents the pole. The control points $(c^x_{i_1 i_2}, c^y_{i_1 i_2})$ are obtained by interpolating the corresponding analytical mapping on the so-called Greville points [23, 24], which are averages of the knots generally lying near the parameter values corresponding to a maximum of the basis functions:

$$\xi_i = \frac{1}{p}(t_{i+1} + t_{i+2} + \cdots + t_{i+p})\,,$$

where we denoted by $(t_0, t_1, \ldots, t_m)$ the knots associated to a generic spline basis of degree p. We note that all the control points at $i_1 = 1$ are equal: $(c^x_{1 i_2}, c^y_{1 i_2}) = (\bar{x}, \bar{y})$. Moreover, (4) implies that $(x(0,\theta), y(0,\theta)) = (\bar{x}, \bar{y})$ for all θ: the whole $s = 0$ line in the logical domain collapses to the pole and the mapping takes a unique value there. From a geometric point of view, a uniform rectangular logical domain is mapped through the mapping $\boldsymbol{F}$ to a polar domain, with one edge collapsing to the pole. $\boldsymbol{F}$ is such that C^k smooth splines in the logical domain remain C^k smooth everywhere except at the pole when mapped onto the polar domain.

4. Semi-Lagrangian advection solver

We now consider the 2D hyperbolic advection equation in (1)

$$\frac{\partial \rho}{\partial t} - E^y \frac{\partial \rho}{\partial x} + E^x \frac{\partial \rho}{\partial y} = 0 \tag{5}$$

and denote by $\boldsymbol{A} = (A^x, A^y) = (-E^y, E^x)$ the 2D advection fields, for simplicity. These are assumed to be well-behaved on the physical Cartesian domain Ω. However, at the discrete level they are represented by splines and need therefore to be evaluated in the logical coordinates (s,θ). We now discuss the optimal choice of coordinates for the backward tracing of the characteristic trajectories associated to (5). It is natural to think of integrating the characteristic equations in either Cartesian or logical coordinates. Therefore, we first discuss these two choices in the following. The characteristic equations in Cartesian coordinates (x, y) read

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} A^x(t, s(x,y), \theta(x,y)) \\ A^y(t, s(x,y), \theta(x,y)) \end{bmatrix}.$$

This is well defined everywhere in the domain, but becomes computationally expensive if the mapping $(s,\theta) \mapsto (x,y)$ is not easy to invert. The characteristic equations in logical coordinates (s,θ) are

$$\begin{bmatrix} \dot{s} \\ \dot{\theta} \end{bmatrix} = J^{-1} \begin{bmatrix} A^x(t, s, \theta) \\ A^y(t, s, \theta) \end{bmatrix}, \tag{6}$$

where J^{-1} is the Jacobian of the inverse mapping,

$$J^{-1} = \frac{J^C}{\det J},$$

with J^C being the transpose of the cofactor matrix of J. The characteristic equations (6) are not defined at the pole, because J^{-1} is singular there. We then suggest to introduce new coordinates (X, Y),

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} s\cos\theta \\ s\sin\theta \end{bmatrix},$$

which we name pseudo-Cartesian coordinates (Figure 2). For a circular mapping, pseudo-Cartesian coordinates reduce to the standard Cartesian ones. The Jacobian of the new transformation $(s, \theta) \mapsto (X, Y)$ is simply

$$\mathcal{J} = \begin{bmatrix} \cos\theta & -s\sin\theta \\ \sin\theta & s\cos\theta \end{bmatrix}.$$

The characteristic equations in pseudo-Cartesian coordinates (X, Y) are

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = (J\mathcal{J}^{-1})^{-1} \begin{bmatrix} A^x(t, s(X,Y), \theta(X,Y)) \\ A^y(t, s(X,Y), \theta(X,Y)) \end{bmatrix}. \tag{7}$$

We first note that the mapping $(s, \theta) \mapsto (X, Y)$ is easier to invert than the original mapping $(s, \theta) \mapsto (x, y)$:

$$\begin{bmatrix} s \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{X^2 + Y^2} \\ \mathrm{atan2}(Y, X) \end{bmatrix},$$

where $\mathrm{atan2}(Y, X)$ returns the principal value of the argument function applied to the complex number $X + iY$ in the range $(-\pi, \pi]$ (which must then be shifted appropriately to the domain interval $[0, 2\pi)$). Moreover, the product of Jacobian matrices appearing in (7) turns out to be well-behaved in the whole logical domain, including the pole. More precisely, the singularity coming from the inverse Jacobian

$$\mathcal{J}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\dfrac{1}{s}\sin\theta & \dfrac{1}{s}\cos\theta \end{bmatrix}$$

when $s \to 0$ is cancelled by the matrix elements of J. The product $J\mathcal{J}^{-1}$ in general looks like

$$J\mathcal{J}^{-1} = \begin{bmatrix} \dfrac{\partial x}{\partial s}\cos\theta - \dfrac{1}{s}\dfrac{\partial x}{\partial \theta}\sin\theta & \dfrac{\partial x}{\partial s}\sin\theta + \dfrac{1}{s}\dfrac{\partial x}{\partial \theta}\cos\theta \\ \dfrac{\partial y}{\partial s}\cos\theta - \dfrac{1}{s}\dfrac{\partial y}{\partial \theta}\sin\theta & \dfrac{\partial y}{\partial s}\sin\theta + \dfrac{1}{s}\dfrac{\partial y}{\partial \theta}\cos\theta \end{bmatrix}. \tag{8}$$

From an analytical point of view, (8) holds for all values of s except at the pole $s = 0$. From a numerical point of view, (8) holds for all values of s sufficiently far from the pole, as far as the factor $1/s$ does not become too large. Therefore, we assume that (8) holds for $s \geq \epsilon$, for a given small ϵ. For $s = 0$ we note that the products $\dfrac{1}{s}\dfrac{\partial x}{\partial \theta}$ and $\dfrac{1}{s}\dfrac{\partial y}{\partial \theta}$ turn out to be well-behaved. More precisely, we have

$$\lim_{s\to 0} \frac{\partial x}{\partial \theta} = 0 \quad \text{and} \quad \lim_{s\to 0} \frac{\partial y}{\partial \theta} = 0,$$

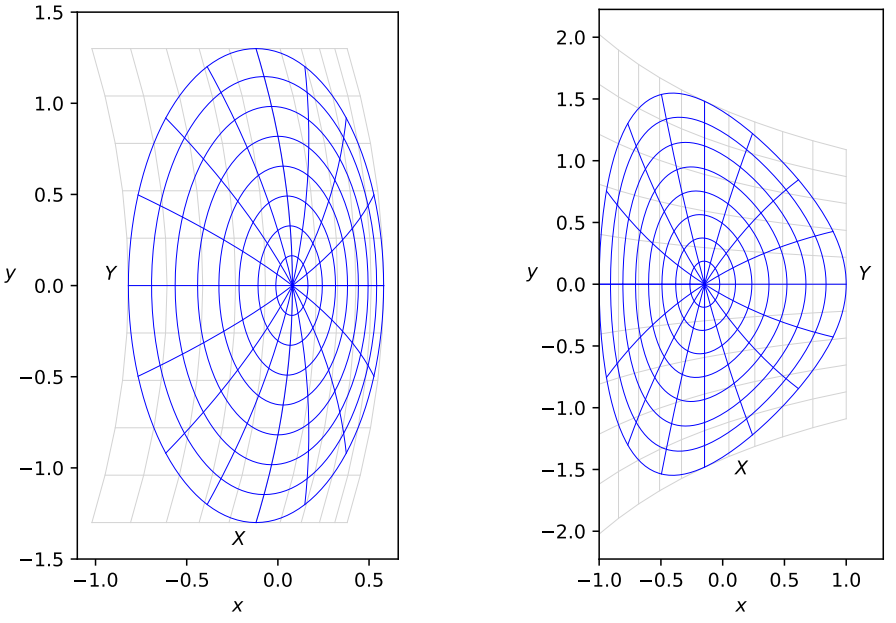

Fig. 2: Pseudo-Cartesian coordinates: the light-grey grids represent the grids in the pseudo-Cartesian coordinates (X, Y) for the mappings (2) (left) and (3) (right), initialized with the parameters in Tables 1 and 2, respectively.

and, expanding in s around the pole,

$$\frac{\partial x}{\partial \theta} = s\,\frac{\partial^2 x}{\partial s\,\partial \theta}(0,\theta) + O(s^2)\,,$$
$$\frac{\partial y}{\partial \theta} = s\,\frac{\partial^2 y}{\partial s\,\partial \theta}(0,\theta) + O(s^2)\,,$$

we obtain

$$\lim_{s\to 0}\frac{1}{s}\frac{\partial x}{\partial \theta} = \frac{\partial^2 x}{\partial s\,\partial \theta}(0,\theta)\,,$$
$$\lim_{s\to 0}\frac{1}{s}\frac{\partial y}{\partial \theta} = \frac{\partial^2 y}{\partial s\,\partial \theta}(0,\theta)\,.$$

Therefore, the matrix elements of $J\mathcal{J}^{-1}$ at the pole take the values

$$J\mathcal{J}^{-1}(0,\theta) = \begin{bmatrix} \dfrac{\partial x}{\partial s}(0,\theta)\cos\theta - \dfrac{\partial^2 x}{\partial s\,\partial\theta}(0,\theta)\sin\theta & \dfrac{\partial x}{\partial s}(0,\theta)\sin\theta + \dfrac{\partial^2 x}{\partial s\,\partial\theta}(0,\theta)\cos\theta \\ \dfrac{\partial y}{\partial s}(0,\theta)\cos\theta - \dfrac{\partial^2 y}{\partial s\,\partial\theta}(0,\theta)\sin\theta & \dfrac{\partial y}{\partial s}(0,\theta)\sin\theta + \dfrac{\partial^2 y}{\partial s\,\partial\theta}(0,\theta)\cos\theta \end{bmatrix}.$$

For the mapping (2) we get

$$(J\mathcal{J}^{-1})^{-1}(0,\theta) = \begin{bmatrix} \dfrac{1+\kappa}{1-\kappa^2} & 0 \\ 0 & \dfrac{1-\kappa}{1-\kappa^2} \end{bmatrix},$$

and for the mapping (3)

$$(J\mathcal{J}^{-1})^{-1}(0,\theta) = \begin{bmatrix} -\sqrt{1+\varepsilon^2} & 0 \\ 0 & \dfrac{2-\sqrt{1+\varepsilon^2}}{e\,\xi} \end{bmatrix}.$$

To connect the two approaches in a smooth way, for $0 < s < \epsilon$ we interpolate linearly the value at the pole and the value at $s = \epsilon$:

$$(J\mathcal{J}^{-1})^{-1}(s,\theta) = \left(1 - \frac{s}{\epsilon}\right)(J\mathcal{J}^{-1})^{-1}(0,\theta) + \frac{s}{\epsilon}(J\mathcal{J}^{-1})^{-1}(\epsilon,\theta)\,.$$

The parameter ϵ can be chosen arbitrarily small, as far as it is consistent with the overall numerical accuracy of our scheme. For the numerical tests discussed in this work we always set $\epsilon = 10^{-12}$.

4.1. Numerical tests

We test the advection solver for a 2D rotating advection field

$$A^x = \omega(y_c - y)\,,$$
$$A^y = \omega(x - x_c)\,,$$

with $\omega = 2\pi$ and $(x_c, y_c) = (0.25, 0.0)$. For a given time step Δt, the analytical flow field reads

$$x(t + \Delta t) = x_c + (x(t) - x_c)\cos(\omega\Delta t) - (y(t) - y_c)\sin(\omega\Delta t)\,,$$
$$y(t + \Delta t) = y_c + (x(t) - x_c)\sin(\omega\Delta t) + (y(t) - y_c)\cos(\omega\Delta t)\,.$$

The numerical solution is compared to the analytical one obtained from this flow field. The initial condition is set to a superposition of cosine bells with elliptical cross sections:

$$\rho(t = 0, x, y) = \frac{1}{2}\Big[\mathcal{G}(r_1(x,y)) + \mathcal{G}(r_2(x,y))\Big],$$

with $\mathcal{G}(r)$ defined by

$$\mathcal{G}(r) = \begin{cases} \cos\left(\dfrac{\pi r}{2a}\right)^4 & r < a \\ 0 & \text{elsewhere} \end{cases},$$

and $r_1(x,y)$ and $r_2(x,y)$ defined by

$$r_1(x,y) = \sqrt{(x - x_0)^2 + 8(y - y_0)^2}\,,$$
$$r_2(x,y) = \sqrt{8(x - x_0)^2 + (y - y_0)^2}\,.$$

Standard tensor-product spline interpolation with no smoothness constraints at the pole is employed. We test the solver on the mapping (3) initialized with the parameters in Table 2 and with $a = 0.3$.

A measure of the error is obtained by taking the maximum in time of the spatial L^2-norm of the numerical error, i.e. the discrepancy between the numerical solution and the analytical one at each mesh location:

$$\mathcal{E} = \max_t \sqrt{\iint \mathrm{d}x\,\mathrm{d}y\,[\rho(t,x,y) - \rho_{\mathrm{ex}}(t,x,y)]^2}\,. \tag{9}$$

Tabel 3 shows the convergence of the scheme while decreasing the time step and keeping the CFL number constant using a 3rd order Runge-Kutta method for the time integration of the characteristics and cubic splines.

We note that there are no order reduction effects due to the singularity. Standard tensor-product spline interpolation with no smoothness constraints at the pole turns out to work fine in the presence of analytical advection fields, provided our choice of coordinates for the time integration of the characteristic trajectories.

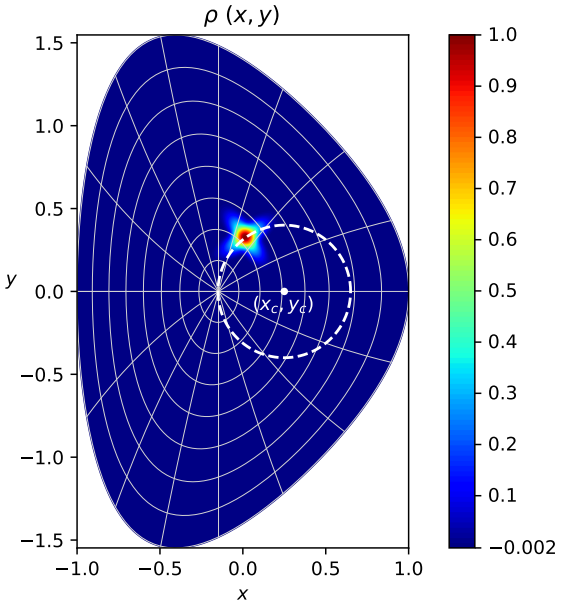

Fig. 3: Numerical test of the advection solver: contour plot of the density ρ at a given time.

Time step Δt	Mesh $N_s \times N_\theta$	Error (L^2)	Order	Error (L^∞)	Order
0.1	64×128	3.20×10^{-2}		3.53×10^{-1}	
0.1/2	128×256	4.06×10^{-3}	2.98	4.34×10^{-2}	3.02
0.1/4	256×512	5.08×10^{-4}	3.00	5.09×10^{-3}	3.09
0.1/8	512×1024	6.37×10^{-5}	3.00	6.13×10^{-4}	3.05
0.1/16	1024×2048	7.97×10^{-6}	3.00	7.52×10^{-5}	3.03

Table 3: Convergence of the advection solver using a 3rd order Runge-Kutta method for the integration of the characteristics and cubic splines.

5. Finite element Poisson solver

We now come to the 2D elliptic Poisson equation in (1)

$$-\Delta\phi = \rho\,. \tag{10}$$

As already pointed out, we are going to solve this equation with a finite element method based on B-splines. Following an isogeometric analysis approach, the same spline basis used to construct the discrete spline mappings will be used as a basis for our finite element method. We recall that we want the potential ϕ returned by our elliptic solver to be at least of class C^1 in the neighborhood of the pole, in order to then get by means of derivatives advection fields for the transport of ρ which are at least continuous at the pole. This is achieved by imposing appropriate C^1 smoothness constraints on the spline basis while solving the linear system obtained from the weak form of (10). A systematic approach to define a set of globally C^1 smooth spline basis functions on singular disk-like domains was developed in

[1] and we will now recall its basic ideas in the following ([1] actually suggested a more general procedure valid also if a higher degree of smoothness is required).

5.1. C^1 smooth polar basis

The idea is to impose the C^0 and C^1 requirements by imposing appropriate constraints on the degrees of freedom corresponding to $i_1 = 1, 2$ for all i_2. More precisely, the basis functions corresponding to these degrees of freedom are replaced by three basis functions, properly defined as linear combinations of the existing ones. In order to guarantee the properties of partition of unity and positivity, [1] suggests to use barycentric coordinates to construct these linear combinations. Taking a triangle enclosing the pole and the first row of control points $(c^x_{2\,i_2}, c^y_{2\,i_2})$, with vertices

$$
\begin{aligned}
V_1 &= (\bar{x} + \tau, \bar{y})\,, \\
V_2 &= \left(\bar{x} - \frac{\tau}{2}, \bar{y} + \frac{\sqrt{3}}{2}\tau\right), \\
V_3 &= \left(\bar{x} - \frac{\tau}{2}, \bar{y} - \frac{\sqrt{3}}{2}\tau\right),
\end{aligned}
$$

with τ given by

$$
\tau = \max\left[\max_{i_2}(-2(c^x_{2\,i_2} - \bar{x})), \max_{i_2}((c^x_{2\,i_2} - \bar{x}) - \sqrt{3}(c^y_{2\,i_2} - \bar{y})), \max_{i_2}((c^x_{2\,i_2} - \bar{x}) + \sqrt{3}(c^y_{2\,i_2} - \bar{y}))\right],
$$

we denote by $(\lambda_1, \lambda_2, \lambda_3)$ the barycentric coordinates of any point with respect to the vertices of this triangle:

$$
\begin{aligned}
\lambda_1(x, y) &= \frac{1}{3} + \frac{2}{3}\frac{1}{\tau}(x - \bar{x})\,, \\
\lambda_2(x, y) &= \frac{1}{3} - \frac{1}{3}\frac{1}{\tau}(x - \bar{x}) + \frac{\sqrt{3}}{3}\frac{1}{\tau}(y - \bar{y})\,, \\
\lambda_3(x, y) &= \frac{1}{3} - \frac{1}{3}\frac{1}{\tau}(x - \bar{x}) - \frac{\sqrt{3}}{3}\frac{1}{\tau}(y - \bar{y})\,.
\end{aligned}
$$

Then, the three new basis functions are defined as

$$
\mathcal{B}_l(s, \theta) = \sum_{i_1=1}^{2}\sum_{i_2=1}^{n_2} e^l_{i_1 i_2} B^s_{i_1}(s) B^\theta_{i_2}(\theta)\,,
$$

with $l = 1, 2, 3$, and the coefficients $e^l_{1\,i_2}$ and $e^l_{2\,i_2}$ given by

$$
e^l_{1\,i_2} = \lambda_l(\bar{x}, \bar{y})\,, \quad e^l_{2\,i_2} = \lambda_l(c^x_{2\,i_2}, c^y_{2\,i_2})\,.
$$

5.2. Finite element solver

We now consider a more general version of Poisson's equation (10) including a finite set of point charges, denoted with the label l, of charges q_l and positions (x_l, y_l):

$$
-\Delta\phi = \rho + \sum_l q_l\,\delta(x - x_l)\delta(y - y_l)\,. \tag{11}
$$

Moreover, we assume that ϕ satisfies homogeneous Dirichlet boundary conditions at the boundary of the Cartesian domain Ω, namely $\phi(\partial\Omega) = 0$. The weak form of (11) is

$$
\int_\Omega \mathrm{d}\Omega\,\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}B_{j_1 j_2} = \int_\Omega \mathrm{d}\Omega\,\rho\,B_{j_1 j_2} + \sum_l q_l B_{j_1 j_2}(s_l, \theta_l) \qquad \forall\, j_1, j_2\,,
$$

with $(s_l, \theta_l) = \boldsymbol{F}^{-1}(x_l, y_l)$. We first expand ϕ and ρ on the basis functions $B_{i_1 i_2}$ as

$$\phi = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \phi_{i_1 i_2} B_{i_1 i_2}\,, \quad \rho = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \rho_{i_1 i_2} B_{i_1 i_2}\,,$$

obtaining

$$\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \phi_{i_1 i_2} \int_\Omega \mathrm{d}\Omega\, \boldsymbol{\nabla} B_{i_1 i_2} \cdot \boldsymbol{\nabla} B_{j_1 j_2} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \rho_{i_1 i_2} \int_\Omega \mathrm{d}\Omega\, B_{i_1 i_2}\, B_{j_1 j_2} + \sum_l q_l B_{j_1 j_2}(s_l, \theta_l) \qquad \forall\, j_1, j_2\,.$$

We now introduce the stiffness and mass tensors

$$\begin{aligned} S_{i_1 i_2 j_1 j_2} &= \int_\Omega \mathrm{d}\Omega\, \boldsymbol{\nabla} B_{j_1 j_2} \cdot \boldsymbol{\nabla} B_{i_1 i_2}\,, \\ M_{i_1 i_2 j_1 j_2} &= \int_\Omega \mathrm{d}\Omega\, B_{j_1 j_2}\, B_{i_1 i_2}\,, \end{aligned}$$

and rewrite our equation as

$$\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} S_{j_1 j_2 i_1 i_2}\, \phi_{i_1 i_2} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} M_{j_1 j_2 i_1 i_2}\, \rho_{i_1 i_2} + \sum_l q_l B_{j_1 j_2}(s_l, \theta_l) \qquad \forall\, j_1, j_2\,. \tag{12}$$

For each finite element we introduce $1 + p_1$ Gauss-Legendre quadrature points and weights along s and $1 + p_2$ Gauss-Legendre quadrature points and weights along θ:

$$\begin{aligned} &(s_1, w_1^s), \ldots, (s_{1+p_1}, w_{1+p_1}^s)\,, \\ &(\theta_1, w_1^\theta), \ldots, (\theta_{1+p_2}, w_{1+p_2}^\theta)\,, \end{aligned} \tag{13}$$

and perform the integrations above on each finite element (superscript e) in the logical domain as

$$\begin{aligned} S^e_{i_1 i_2 j_1 j_2} &= \sum_{q_1=1}^{1+p_1} \sum_{q_2=1}^{1+p_2} \sqrt{\det G(s_{q_1}, \theta_{q_2})}\, w_{q_1}^s w_{q_2}^\theta \begin{pmatrix} \dfrac{\partial}{\partial s} B_{j_1 j_2}(s_{q_1}, \theta_{q_2}) \\ \dfrac{\partial}{\partial \theta} B_{j_1 j_2}(s_{q_1}, \theta_{q_2}) \end{pmatrix}^T G^{-1}(s_{q_1}, \theta_{q_2}) \begin{pmatrix} \dfrac{\partial}{\partial s} B_{i_1 i_2}(s_{q_1}, \theta_{q_2}) \\ \dfrac{\partial}{\partial \theta} B_{i_1 i_2}(s_{q_1}, \theta_{q_2}) \end{pmatrix}, \\ M^e_{i_1 i_2 j_1 j_2} &= \sum_{q_1=1}^{1+p_1} \sum_{q_2=1}^{1+p_2} \sqrt{\det G(s_{q_1}, \theta_{q_2})}\, w_{q_1}^s w_{q_2}^\theta B_{j_1 j_2}(s_{q_1}, \theta_{q_2})\, B_{i_1 i_2}(s_{q_1}, \theta_{q_2})\,, \end{aligned}$$

where G is the metric matrix. Equation (12) can then be written in matrix form by using a multi-index notation. More precisely, defining the multi-indices $I = (i_1, i_2)$ and $J = (j_1, j_2)$ we write (12) as

$$\sum_I S_{JI}\, \phi_I = \sum_I M_{JI}\, \rho_I + \sum_l q_l B_J(s_l, \theta_l) \qquad \forall J\,,$$

or in the more compact matrix form

$$\mathcal{S} \cdot \boldsymbol{\phi} = \mathcal{M} \cdot \boldsymbol{\rho} + \sum_l q_l\, \boldsymbol{b}(s_l, \theta_l)\,, \tag{14}$$

where we introduced the matrices $\mathcal{S}$ and $\mathcal{M}$ with elements S_{JI} and M_{JI}, respectively, the vectors $\boldsymbol{\phi}$, $\boldsymbol{\rho}$ and $\boldsymbol{b}(s_l, \theta_l)$ with elements ϕ_I, ρ_I and $B_J(s_l, \theta_l)$, respectively, and the $\cdot$ product denoting the usual matrix-vector multiplication.

The C^1 smoothness constraint is imposed by projecting the matrices and vectors in (14) onto the sub-space of C^1 smooth polar splines by applying

$$\mathcal{P} = \begin{pmatrix} L & 0 \\ 0 & I \end{pmatrix},$$

where L contains the barycentric coordinates of the control points. More precisely, the stiffness and mass matrices $\mathcal{S}$ and $\mathcal{M}$ are projected to $\mathcal{S}^* = \mathcal{P}^T \mathcal{S} \mathcal{P}$ and $\mathcal{M}^* = \mathcal{P}^T \mathcal{M} \mathcal{P}$, respectively, and the vectors $\boldsymbol{\phi}$, $\boldsymbol{\rho}$ and $\boldsymbol{b}(s_l, \theta_l)$ are projected to $\boldsymbol{\phi}^* = \mathcal{P}^T \boldsymbol{\phi}$, $\boldsymbol{\rho}^* = \mathcal{P}^T \boldsymbol{\rho}$ and $\boldsymbol{b}^*(s_l, \theta_l) = \mathcal{P}^T \boldsymbol{b}(s_l, \theta_l)$, respectively. Therefore, (14) reads

$$\mathcal{S}^* \cdot \boldsymbol{\phi}^* = \mathcal{M}^* \cdot \boldsymbol{\rho}^* + \sum_l q_l \, \boldsymbol{b}^*(s_l, \theta_l) \,. \tag{15}$$

The linear system (15) is then solved with an iterative conjugate gradient method [25, 26] and the resulting solution is projected back onto the tensor-product space. More details about the implementation of the solver are discussed in Appendix A.

5.3. Numerical tests

We first test the Poisson solver on a circular mapping looking for a solution ϕ_{ex} of the form

$$\phi_{\text{ex}}(x, y) = (1 - x^2 - y^2) \cos(2\pi x) \sin(2\pi y) \,,$$

which satisfies homogeneous Dirichlet boundary conditions at $s = \sqrt{x^2 + y^2} = 1$ and corresponds to a density ρ of the form

$$\rho(x, y) = 4[2\pi^2(1 - x^2 - y^2) + 1] \cos(2\pi x) \sin(2\pi y) - 8\pi[x \sin(2\pi x) \sin(2\pi y) - y \cos(2\pi x) \cos(2\pi y)] \,.$$

A measure of the error is obtained by computing the spatial L^2-norm of the numerical error, i.e. the discrepancy between the numerical solution $\phi(x, y)$ and the exact one $\phi_{\text{ex}}(x, y)$ at each mesh location:

$$\mathcal{E} = \sqrt{\iint \mathrm{d}x \, \mathrm{d}y \, [\phi(x, y) - \phi_{\text{ex}}(x, y)]^2} \,.$$

Table 4 shows the convergence of the solver while increasing the mesh size using cubic splines.

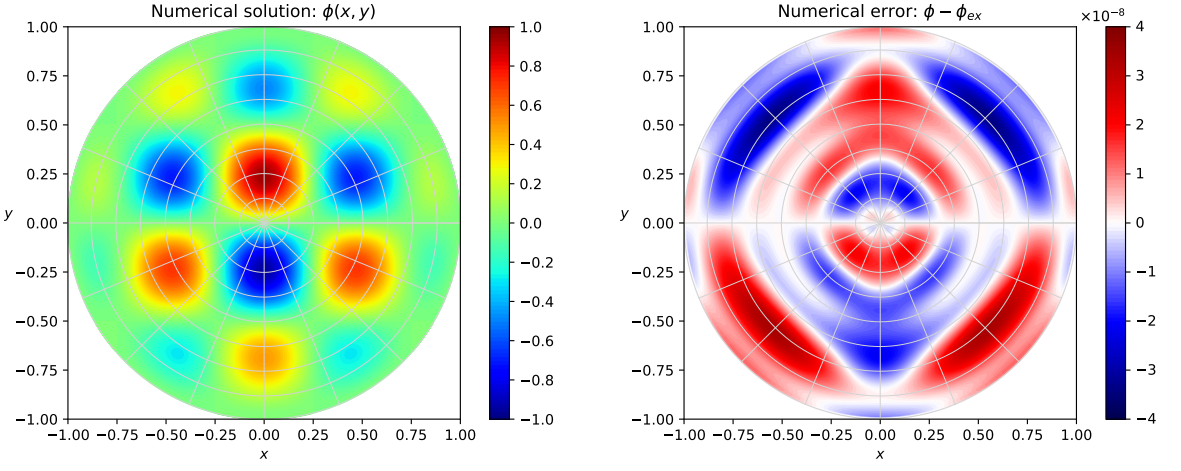

Fig. 4: Numerical test of the Poisson solver on a circular mapping: solution (left) and numerical error (right).

Mesh $N_s \times N_\theta$	Error (L^2)	Order	Error (L^∞)	Order
32×64	7.78×10^{-6}		1.24×10^{-5}	
64×128	3.91×10^{-7}	4.31	5.95×10^{-7}	4.38
128×256	2.22×10^{-8}	4.14	3.36×10^{-8}	4.15
256×512	1.33×10^{-9}	4.06	2.00×10^{-9}	4.07
512×1024	8.12×10^{-11}	4.03	1.22×10^{-10}	4.04

Table 4: Convergence of the Poisson solver on a circular mapping using cubic splines.

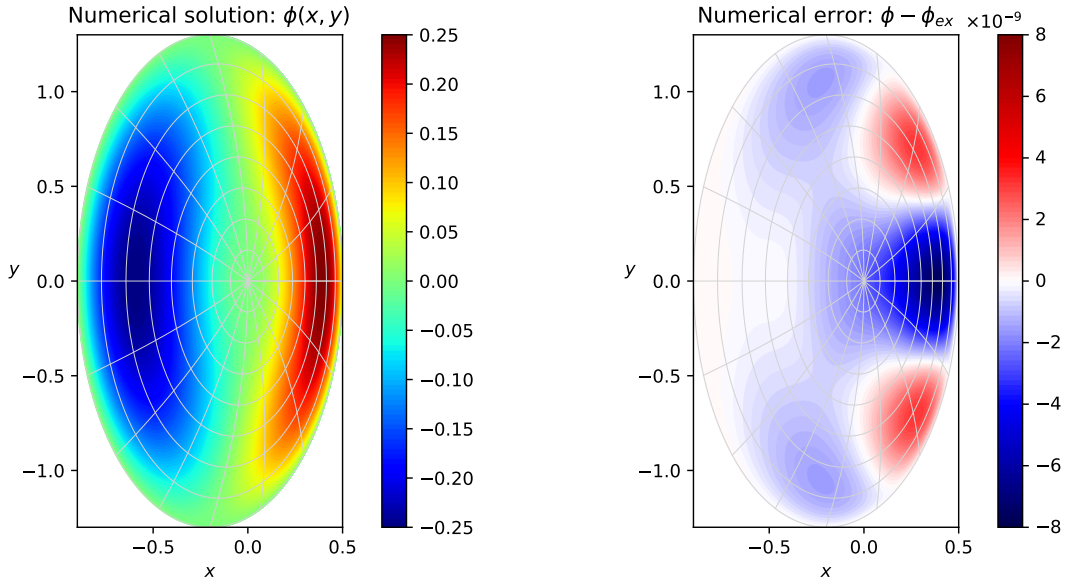

Fig. 5: Numerical test of the Poisson solver on the mapping (2) initialized with the parameters in Table 1: solution (left) and numerical error (right).

We then test the solver on the mapping (2) looking for a separable solution ϕ_{ex} of the form

$$\phi_{\text{ex}}(s, \theta) = s^2(1 - s^2)\cos(\theta)\,,$$

which satisfies homogeneous Dirichlet boundary conditions at $s = 1$. Table 5 shows again the convergence of the solver while increasing the mesh size using cubic splines.

5.4. Evaluation of the electric field

The advection fields for the transport of ρ are obtained from the potential ϕ by means of derivatives. This section suggests a strategy to evaluate the gradient of ϕ while taking into account the singularity at the pole. We first denote by $\widehat{\phi} = \phi \circ \boldsymbol{F}$ the composition of ϕ with the mapping (i.e. $\widehat{\phi}$ must be evaluated in the logical domain) and $\widehat{\phi}_{i_1 i_2}$ the coefficients of its spline representation:

$$\widehat{\phi}(s, \theta) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \widehat{\phi}_{i_1 i_2} B^s_{i_1}(s) B^\theta_{i_2}(\theta)\,.$$

Mesh $N_s \times N_\theta$	Error (L^2)	Order	Error (L^∞)	Order
32×64	8.17×10^{-7}		2.18×10^{-6}	
64×128	4.71×10^{-8}	4.12	1.25×10^{-7}	4.12
128×256	2.85×10^{-9}	4.05	7.55×10^{-9}	4.05
256×512	1.75×10^{-10}	4.03	4.65×10^{-10}	4.02
512×1024	1.09×10^{-11}	4.01	2.88×10^{-11}	4.01

Table 5: Convergence of the Poisson solver on the mapping (2) initialized with the parameters in Table 1 using cubic splines.

The components of its gradient in the logical domain $\widehat{\boldsymbol{\nabla}\phi} = \left(\frac{\partial\widehat{\phi}}{\partial s}, \frac{\partial\widehat{\phi}}{\partial\theta}\right)^T$ are given by

$$\frac{\partial\widehat{\phi}}{\partial s}(s,\theta) = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\widehat{\phi}_{i_1 i_2} B^{s\prime}_{i_1}(s) B^{\theta}_{i_2}(\theta)\,,$$
$$\frac{\partial\widehat{\phi}}{\partial\theta}(s,\theta) = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\widehat{\phi}_{i_1 i_2} B^{s}_{i_1}(s) B^{\theta\prime}_{i_2}(\theta)\,.$$

Their value at the pole $s = 0$ is

$$\frac{\partial\widehat{\phi}}{\partial s}(0,\theta) = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\widehat{\phi}_{i_1 i_2} B^{s\prime}_{i_1}(0) B^{\theta}_{i_2}(\theta) = B^{s\prime}_1(0)\sum_{i_2=1}^{n_2}(\widehat{\phi}_{1\,i_2} - \widehat{\phi}_{2\,i_2}) B^{\theta}_{i_2}(\theta)\,,$$
$$\frac{\partial\widehat{\phi}}{\partial\theta}(0,\theta) = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\widehat{\phi}_{i_1 i_2} B^{s}_{i_1}(0) B^{\theta\prime}_{i_2}(\theta)\,.$$

The physical Cartesian components of the gradient are then obtained from the ones in logical coordinates through the inverse of the Jacobian matrix:

$$\boldsymbol{\nabla}\phi(s,\theta) = (J^{-1})^T(s,\theta)\,\widehat{\boldsymbol{\nabla}\phi}(s,\theta)\,. \tag{16}$$

From an analytical point of view, (16) holds for all values of s except at the pole $s = 0$, where the inverse Jacobian is not defined. From a numerical point of view, (16) holds for all values of s sufficiently far from the pole, as far as the inverse Jacobian does not become too large. Therefore, we assume that (16) holds for $s \geq \epsilon$, for a given small ϵ. For $s = 0$ all the information is contained in the partial derivative of $\widehat{\phi}$ with respect to s, while the partial derivative with respect to θ goes to zero. The partial derivative with respect to s takes a different value for each value of θ. Recalling that a partial derivative has the geometrical meaning of a directional derivative along a tangent-basis vector, and given that ϕ is of class C^1, the idea is to combine two given values corresponding to two different values of θ and extract from them the physical Cartesian components of the gradient at the pole. The two chosen values of θ must correspond to linearly independent directions, so that from

$$\frac{\partial\widehat{\phi}}{\partial s}(0,\theta_1) = \boldsymbol{\nabla}\phi\cdot\boldsymbol{e}_s = (\boldsymbol{\nabla}\phi)_x\frac{\partial x}{\partial s}(0,\theta_1) + (\boldsymbol{\nabla}\phi)_y\frac{\partial y}{\partial s}(0,\theta_1)\,,$$
$$\frac{\partial\widehat{\phi}}{\partial s}(0,\theta_2) = \boldsymbol{\nabla}\phi\cdot\boldsymbol{e}_s = (\boldsymbol{\nabla}\phi)_x\frac{\partial x}{\partial s}(0,\theta_2) + (\boldsymbol{\nabla}\phi)_y\frac{\partial y}{\partial s}(0,\theta_2)\,,$$

the two components $(\boldsymbol{\nabla}\phi)_x$ and $(\boldsymbol{\nabla}\phi)_y$ can be obtained. Each possible couple of linearly indepentent values of θ should produce the same result. To connect the two approaches in a smooth way, for $0 < s < \epsilon$ we interpolate linearly the value at the pole and the value at $s = \epsilon$:

$$\boldsymbol{\nabla}\phi(s,\theta) = \left(1 - \frac{s}{\epsilon}\right)\boldsymbol{\nabla}\phi(0,\theta) + \frac{s}{\epsilon}\boldsymbol{\nabla}\phi(\epsilon,\theta)\,.$$

The parameter ϵ can be chosen arbitrarily small, as far as it is consistent with the overall numerical accuracy of our scheme. For the numerical tests discussed in this work we always set $\epsilon = 10^{-12}$.

6. Self-consistent test case: guiding-center model

We now address the solution of the 2D guiding-center model

$$\begin{cases} \dfrac{\partial\rho}{\partial t} - E^y\dfrac{\partial\rho}{\partial x} + E^x\dfrac{\partial\rho}{\partial y} = 0 \\ -\Delta\phi = \rho \end{cases}, \tag{17}$$

which couples the hyperbolic transport equation and Poisson's equation in a self-consistent model. Physical quantities conserved by the model are the total mass and energy

$$\begin{aligned} \mathcal{M}(t) &= \int \mathrm{d}x\,\mathrm{d}y\,\rho(t,x,y)\,, \\ \mathcal{W}(t) &= \int \mathrm{d}x\,\mathrm{d}y\,|\boldsymbol{E}(t,x,y)|^2\,. \end{aligned} \tag{18}$$

These integrals are computed using the Gauss-Legendre quadrature points (13) on each finite element (superscript e) as for the assembly of the stiffness and mass matrices in the finite element solution of Poisson's equation:

$$\begin{aligned} \mathcal{M}^e(t) &= \sum_{q_1=1}^{1+p_1}\sum_{q_2=1}^{1+p_2}\sqrt{\det G(s_{q_1},\theta_{q_2})}\,w^s_{q_1}w^\theta_{q_2}\,\rho(t,s_{q_1},\theta_{q_2})\,, \\ \mathcal{W}^e(t) &= \sum_{q_1=1}^{1+p_1}\sum_{q_2=1}^{1+p_2}\sqrt{\det G(s_{q_1},\theta_{q_2})}\,w^s_{q_1}w^\theta_{q_2}\,|\boldsymbol{E}(t,s_{q_1},\theta_{q_2})|^2\,. \end{aligned} \tag{19}$$

We denote the relative errors for the conservation of the invariants in (18) as $\mathcal{E}_\mathcal{M}$ and $\mathcal{E}_\mathcal{W}$, respectively:

$$\begin{aligned} \mathcal{E}_\mathcal{M}(t) &= \frac{\mathcal{M}(0) - \mathcal{M}(t)}{\mathcal{M}(0)}\,, \\ \mathcal{E}_\mathcal{W}(t) &= \frac{\mathcal{E}(0) - \mathcal{E}(t)}{\mathcal{E}(0)}\,. \end{aligned} \tag{20}$$

Before describing the actual numerical tests we performed on this model, it is necessary to briefly discuss our time-advancing strategy and how we deal with the problem of defining an equilibrium density on complex mappings while initializing our simulations.

6.1. Time integration

The time integrator we decided to use is the second order explicit integrator described in [27]. The strategy is based on a predictor-corrector procedure, where high order temporal accuracy is built up from lower order predictions.

Considering the backward tracing of the characteristic trajectories for the density distribution ρ, denoting by $\boldsymbol{X}$ the 2D grid coordinates and by $\boldsymbol{A}$ the 2D advection fields, a first order prediction is given by

$$\boldsymbol{X}^{(1)} = \boldsymbol{X} - \boldsymbol{A}(\boldsymbol{X})\Delta t\,,$$

where Δt is the chosen time step. From $\boldsymbol{X}^{(1)}$ an intermediate density $\rho^{(1)}$ can be computed by means of interpolation and intermediate advection fields $\boldsymbol{A}^{(1)}$ can be computed from $\rho^{(1)}$ by solving Poisson's equation. Given the first order prediction, a second order accurate scheme is given by

$$\boldsymbol{X}^{(2)} = \boldsymbol{X} - \frac{1}{2}\Big[\boldsymbol{A}(\boldsymbol{X}^{(1)}) + \boldsymbol{A}^{(1)}(\boldsymbol{X})\Big]\Delta t\,.$$

In the presence of point charges (or point-like vortices), this second order scheme reduces to Heun's method (also known as "explicit trapezoidal rule"):

$$\begin{aligned}\boldsymbol{X}^{(1)} &= \boldsymbol{X} + \boldsymbol{A}(\boldsymbol{X})\Delta t\,,\\ \boldsymbol{X}^{(2)} &= \boldsymbol{X} + \frac{1}{2}\Big[\boldsymbol{A}(\boldsymbol{X}) + \boldsymbol{A}^{(1)}(\boldsymbol{X}^{(1)})\Big]\Delta t\,.\end{aligned}$$

The time step should be small enough to resolve the background phase-space flow field and its variation in time.

6.2. Numerical equilibria

As already pointed out, defining an equilibrium density ρ and a corresponding equilibrium potential ϕ for the system (17) becomes non-trivial on complex mappings. In the case of circular mappings, any axisymmetric density function $\rho(s,\theta) = \bar{\rho}(s)$, independent of the angle variable θ, turns out to be an equilibrium for the transport equation in (17). For more complex mappings we follow a numerical procedure suggested by [28] and references therein to compute an equilibrium couple (ρ,ϕ). The equilibrium is determined by the eigenvalue problem of finding $(\phi,\sigma) \in H^1\times\mathbb{R}$ such that $-\Delta\phi = \sigma\, f(\phi)$, with given f such that $f'(\phi) \neq 0$ in some limited domain Ω, with boundary conditions such that the Poisson equation $-\Delta\phi = \rho$, with a fixed source ρ, is well-posed (we consider here only homogeneous Dirichlet boundary conditions $\phi(\partial\Omega) = 0$). Given initial data (ϕ_0,σ_0) we solve for $i \geq 1$

$$\begin{aligned}\rho_i &= \sigma_{i-1} f(\phi_{i-1})\,,\\ \phi_i^* &= -\Delta^{-1}\rho_i\,,\\ \phi_i &= c\,\phi_i^*\,,\\ \sigma_i &= c\,\sigma_{i-1}\,,\end{aligned} \tag{21}$$

iterating over i. The normalization parameter c can be set by choosing either a maximum value $\bar{\phi}$ for ϕ or a maximum value $\bar{\rho}$ for ρ. In the first case c reads $c = \bar{\phi}/\|\phi_i^*\|_\infty$, while in the second case it turns out to satisfy the equation $cf(c\,\|\phi_i^*\|_\infty) = \bar{\rho}/\sigma_{i-1}$. The eigenvalue problem does not have a unique solution, but the algorithm is supposed to converge to the ground state, i.e. the eigenstate with minimum eigenvalue. Figure 6 illustrates for example the equilibrium obtained in this way for $f(\phi) = \phi^2$ with $\bar{\rho} = 1$ on a circular mapping and on the mapping (3) initialized with the parameters in Table 2.

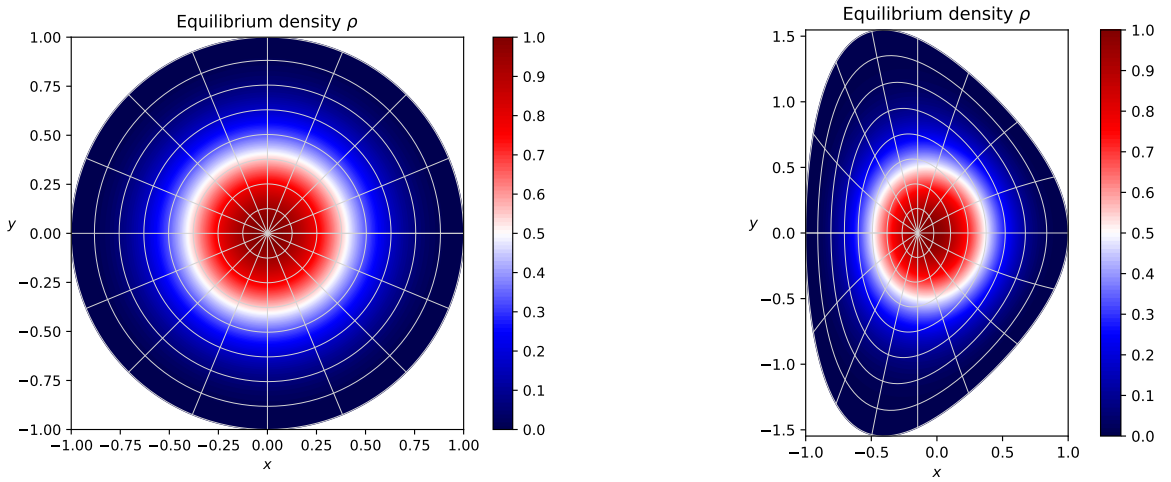

Fig. 6: Numerical equilibrium density ρ obtained with $f(\phi) = \phi^2$ and $\bar{\rho} = 1$ on a circular mapping (left) and on the mapping (3) (right) initialized with the parameters in Table 2.

6.3. Numerical test: diocotron instability

As a first test we investigate the evolution of the diocotron instability in circular geometry. From a physical point of view, this corresponds to studying a non-neutral plasma in cylindrical geometry, where the plasma particles are confined radially by a uniform axial magnetic field with a cylindrical conducting wall located at the outer boundary. Following [4], we consider an initial density profile $\rho(t = 0, r, \theta)$ corresponding to an annular layer, namely

$$\rho(t = 0, r, \theta) = \begin{cases} 1 + \epsilon \cos(m\theta) & r^- \leq r \leq r^+ \\ 0 & \text{elsewhere} \end{cases}, \tag{22}$$

where we denote the logical coordinate s by r, as it corresponds precisely to a standard radial coordinate. A uniform equilibrium density is perturbed with a radial-independent perturbation with azimuthal mode number m and amplitude ϵ. Following the linear dispersion analysis in [4], we obtain the dispersion relation for the complex eigenfrequency ω

$$\left(\frac{\omega}{\omega_D}\right)^2 - b_m \frac{\omega}{\omega_D} + c_m = 0\,, \tag{23}$$

where ω_D is the diocotron frequency ($\omega_D = 1/2$ in our units), and b_m and c_m are given by

$$b_m = m\left[1 - \left(\frac{r^-}{r^+}\right)^2\right] + (r^+)^{2m} - (r^-)^{2m}\,,$$
$$c_m = m\left[1 - \left(\frac{r^-}{r^+}\right)^2\right]\left[1 - (r^-)^{2m}\right] - \left[1 - \left(\frac{r^-}{r^+}\right)^{2m}\right]\left[1 - (r^+)^{2m}\right].$$

If $4c_m > b_m^2$, then the oscillation frequencies resulting from (23) form complex conjugate pairs. The solution with $\mathrm{Im}\,\omega > 0$ corresponds to the diocotron instability and describes how rapidly the fluctuating potential grows, as we write the full electric potential as the sum of an equilibrium time-indepedent potential $\phi_0(r)$ and a fluctuating component $\delta\phi(t, r, \theta)$:

$$\phi(t, r, \theta) = \phi_0(r) + \delta\phi(t, r, \theta)e^{i(m\theta - \omega t)}\,.$$

The quantity of interest in this regard is the L^2-norm of the fluctuating potential

$$\|\phi(t,r,\theta)-\phi_0(r)\|_{L^2} = \sqrt{\int \mathrm{d}x\,\mathrm{d}y\,[\phi(t,r,\theta)-\phi_0(r)]^2}\,,$$

where the integration is again performed on the Gauss-Legendre quadrature points (13). In order to represent the initial density profile in the space of tensor-product splines we modify (22) by a radial smoothing to avoid discontinuities:

$$\rho(t=0,r,\theta) = \begin{cases} [1+\epsilon\cos(m\theta)]\exp\Big[-\Big(\dfrac{r-\bar{r}}{d}\Big)^p\Big] & r^- \le r \le r^+ \\ 0 & \text{elsewhere} \end{cases}, \tag{24}$$

with $\bar{r} := (r^+ + r^-)/2$ and $d = (r^+ - r^-)/2$. If the smoothing layer is small enough, we can still rely on the analytical result obtained for the dispersion relation in the case of the sharp annular layer (22). The numerical results have been verified against the analytical dispersion relation for a perturbation with azimuthal mode number $m = 9$ and amplitude $\epsilon = 10^{-4}$. The numerical growth rate is in good agreement with the analytical one, $\mathrm{Im}\,\omega = 0.17963095941144$, for $20 \lesssim t \lesssim 50$. This time interval corresponds to the linear growth phase. At time $t \approx 50$, the system enters its non-linear phase. The simulation is run with $N_r \times N_\theta = 128 \times 256$ and $\Delta t = 0.002$. Additional parameters defining the initial condition (24) have been set to $r^- = 0.45$, $r^+ = 0.50$ and $p = 50$. Numerical results are illustrated in Figures 7 and 8. For the conservation of mass and energy we get

$$\max_{t\in[0,70]} |\mathcal{E}_{\mathcal{M}}(t)| \approx 1.6\times 10^{-4}\,, \quad \max_{t\in[0,70]} |\mathcal{E}_{\mathcal{W}}(t)| \approx 2.1\times 10^{-4}\,.$$

The time evolution of the relative errors on these conserved quantities is shown in Figure 9.

6.4. Numerical test: vortex merger

In the context of incompressible inviscid 2D Euler fluids, we can observe the merger of two macroscopic vortices by setting up initial conditions qualitatively similar to those described in [5]. We consider an equilibrium obtained with the numerical procedure (21) with $f(\phi) = \phi$ with $\bar{\phi} = 1$ and add on top of that Gaussian perturbations of the form

$$\delta\rho = \epsilon \exp\Big[-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}\Big],$$

with amplitude $\epsilon = 10^{-4}$, width $\sigma = 0.08$ and centered in $(x_0, y_0) = (0.08, -0.14)$ and $(x_0, y_0) = (-0.08, 0.14)$, respectively. The initial perturbation and its evolution in time are shown in Figure 10. The simulation is run with $N_r \times N_\theta = 128 \times 256$ and time step $\Delta t = 0.001$. For the conservation of mass and energy we get

$$\max_{t\in[0,10]} |\mathcal{E}_{\mathcal{M}}(t)| \approx 1.1\times 10^{-7}\,, \quad \max_{t\in[0,10]} |\mathcal{E}_{\mathcal{W}}(t)| \approx 1.3\times 10^{-7}\,.$$

The time evolution of the relative errors on these conserved quantities is shown in Figure 11.

6.5. Numerical test: point-like vortex dynamics

We also investigate the dynamics of point-like vortices (or point charges) on a non-uniform equilibrium, following the discussion in [11]. Our aim is to show that the numerical approaches suggested in this work can be applied straightforwardly in the context of particle-in-cell methods. The examples discussed here can be considered as limit

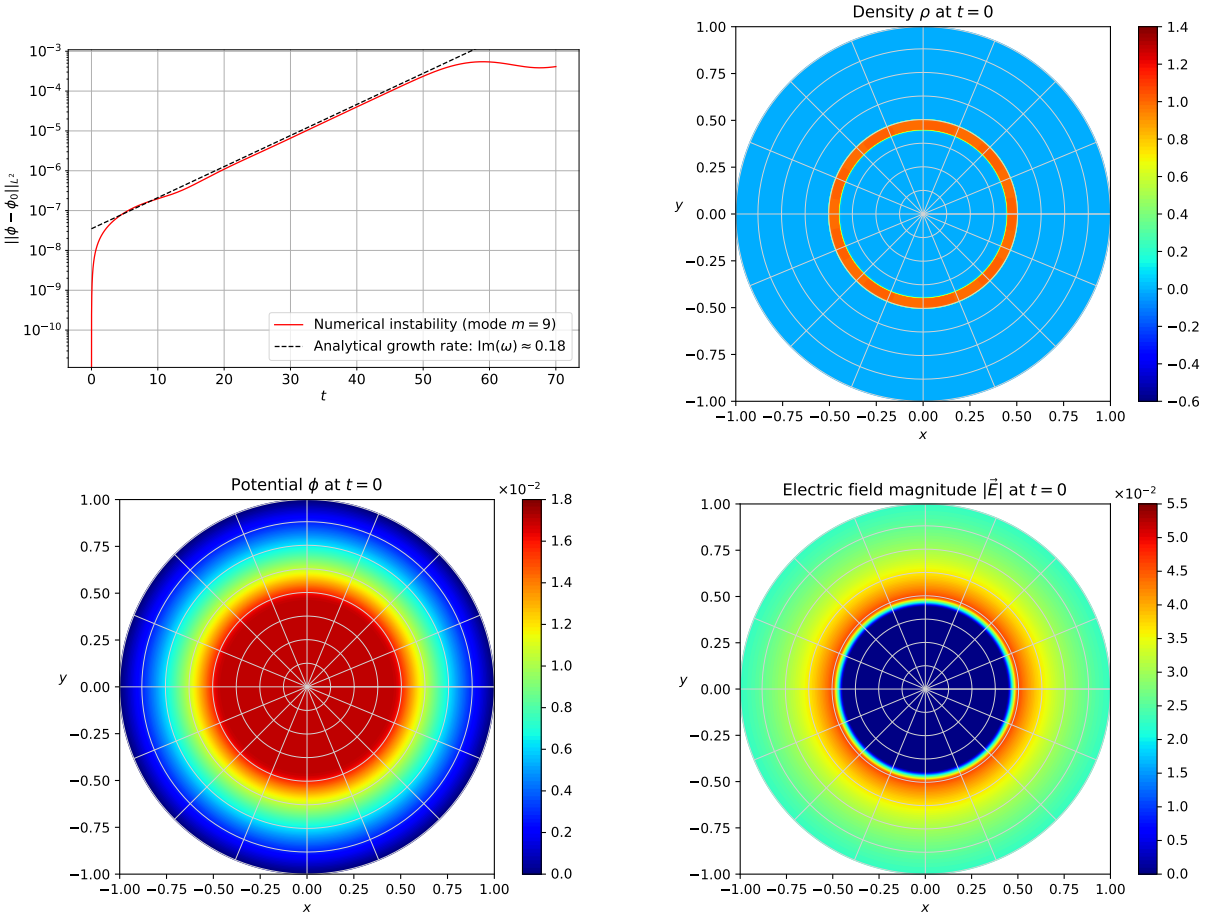

Fig. 7: Numerical simulation of the diocotron instability. From left to right, from top to bottom: L^2-norm of the electric potential, contour plots of density ρ, electric potential ϕ and electric field magnitude $|\boldsymbol{E}|$ at initial time $t = 0$.

cases of usual particle-in-cell simulations, as we will include only one single point-like vortex (or point charge) in the system. Since our strategy turns out to work well for this extreme scenario, we do not expect issues to appear when dealing with the usual case of large numbers of particles. For a circular mapping, we consider an equilibrium density of the form

$$\rho(r) = \begin{cases} 1 - 1.25\, r & r \leq 0.8 \\ 0 & r > 0.8 \end{cases} .$$

Figure 12 shows the local stream lines of the advection field near positive and negative point-like vortices at the initial time in a rotating frame where the point-like vortices are initially at rest. This is obtained in practice by rotating given coordinates (x, y) at time t as

$$x' = x\cos(-\omega t) - y\sin(-\omega t)\,,$$

$$y' = x\sin(-\omega t) + y\cos(-\omega t)\,,$$

where $\omega = 0.3332$ represents the angular velocity of the background. Figure 13 show results for point-like vortices of intensity $q = \pm 0.0025$ at the initial position $r = 0.4$ and $\theta = 0$, again viewed in a rotating frame (time is here normalized as $T = 0.1668\, t$). As explained in [11], positive point-like vortices (clumps) drift transverse to the shear

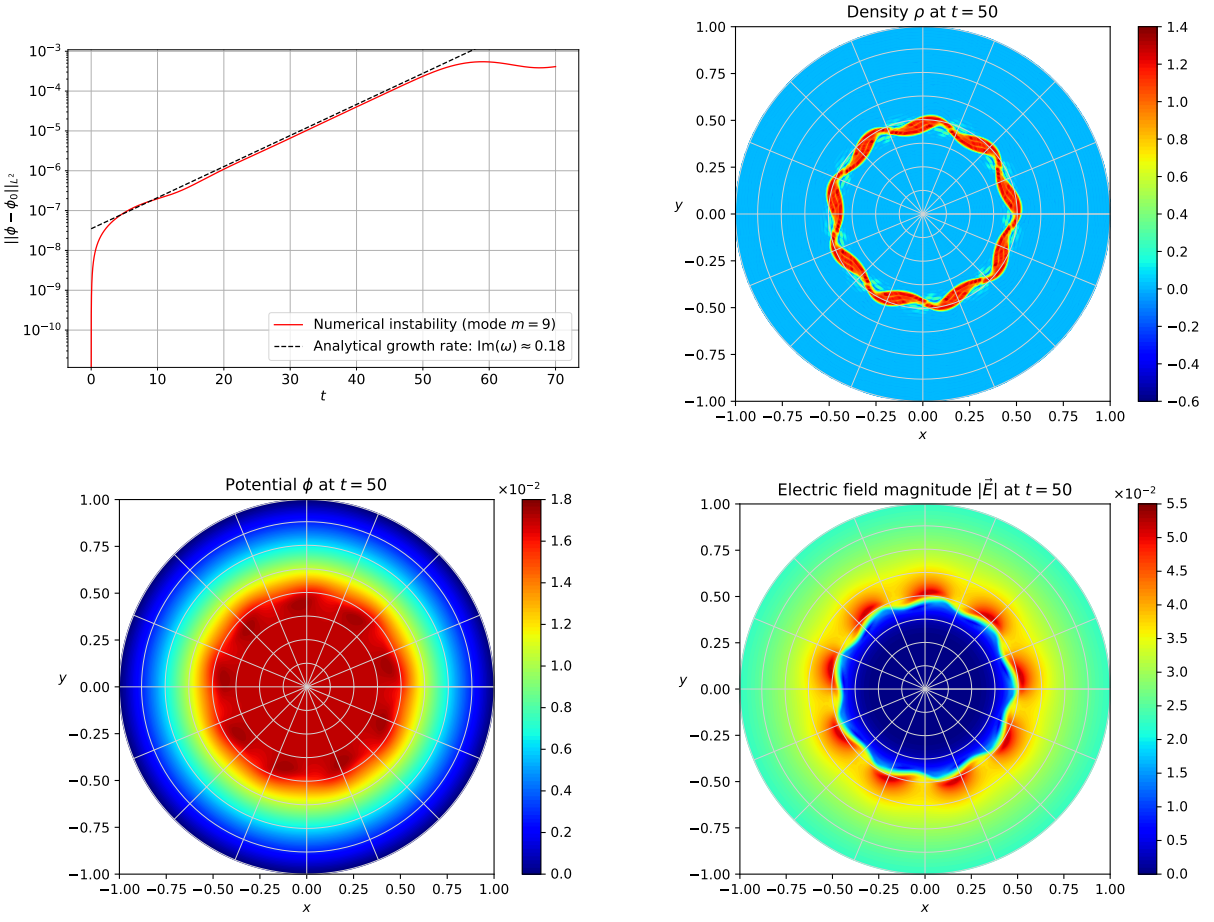

Fig. 8: Numerical simulation of the diocotron instability. From left to right, from top to bottom: L^2-norm of the electric potential, contour plots of density ρ, electric potential ϕ and electric field magnitude $|\boldsymbol{E}|$ at time $t = 50$ (end of the linear phase).

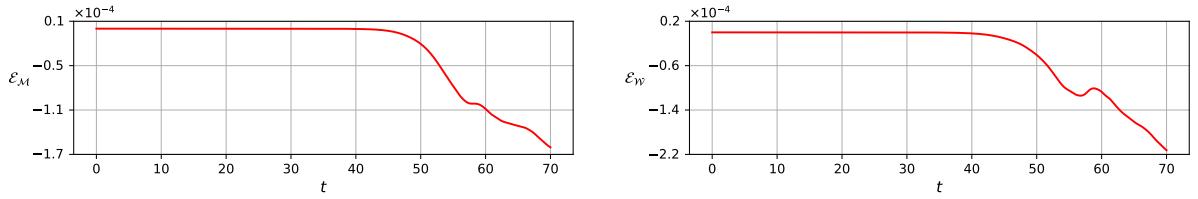

Fig. 9: Numerical simulation of the diocotron instability: time evolution of the relative errors on the total mass (left) and energy (right).

flow, up the background vorticity gradient, while negative point-like vortices (holes) drift down the gradient. Figure 14 shows the time evolution of the vortices' radial position. The simulation is run with $N_r \times N_\theta = 256 \times 512$ and time step $\Delta t = 0.005$. The time step is chosen small enough to resolve the oscillations due to the self-force experienced by the point-like vortices. Special techniques may be used to reduce self-force effects on non-uniform meshes (or even unstructured meshes) [29], but they will not be considered in this work. Our numerical results are in agreement, both

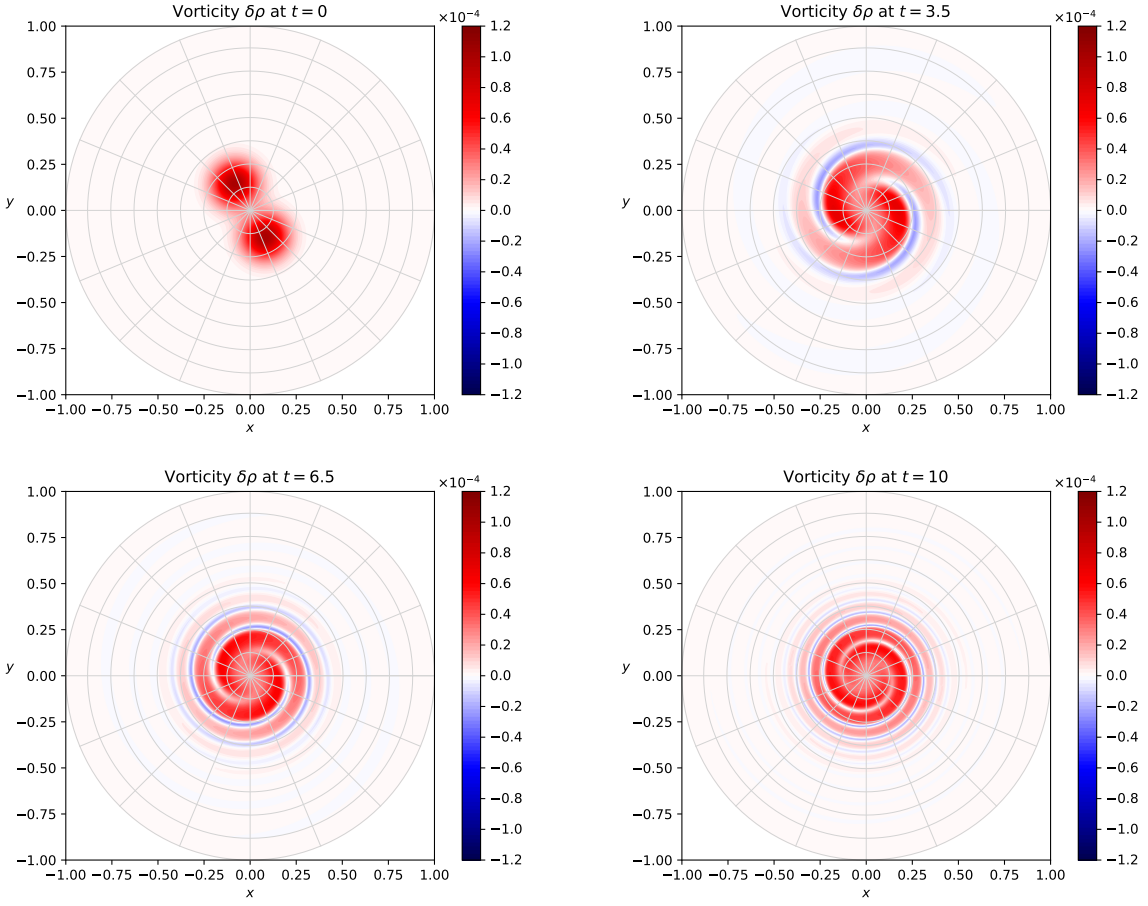

Fig. 10: Vortex merger on a circular mapping: contour plot of the vorticity at different times (numerical equilibrium has been subtracted).

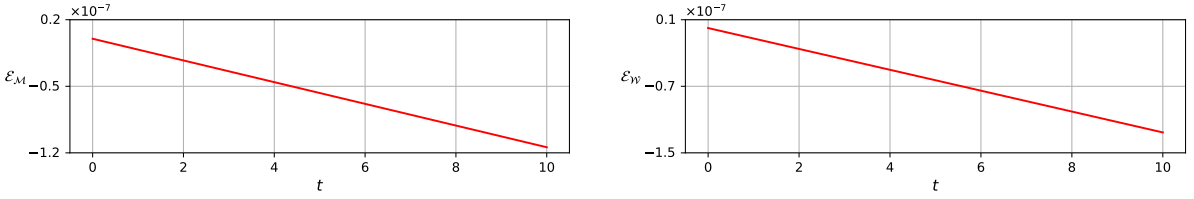

Fig. 11: Vortex merger on a circular mapping: time evolution of the relative errors on the total mass (left) and energy (right).

qualitatively and quantitatively, with the ones discussed in [11]. For the conservation of mass and energy we get

$$\max_{T\in[0,5.838]} |\mathcal{E}_{\mathcal{M}}(T)| \approx 6.9\times 10^{-6}\,, \qquad \max_{T\in[0,5.838]} |\mathcal{E}_{\mathcal{W}}(T)| \approx 8.4\times 10^{-3}$$

for the positive point-like vortex and

$$\max_{T\in[0,5.838]} |\mathcal{E}_{\mathcal{M}}(T)| \approx 6.9\times 10^{-6}\,, \qquad \max_{T\in[0,5.838]} |\mathcal{E}_{\mathcal{W}}(T)| \approx 7.3\times 10^{-3}$$

for the negative point-like vortex. The time evolution of the relative errors on these conserved quantities is shown in Figure 15. Similar simulations on the mapping (3), initialized with an equilibrium obtained with the numerical

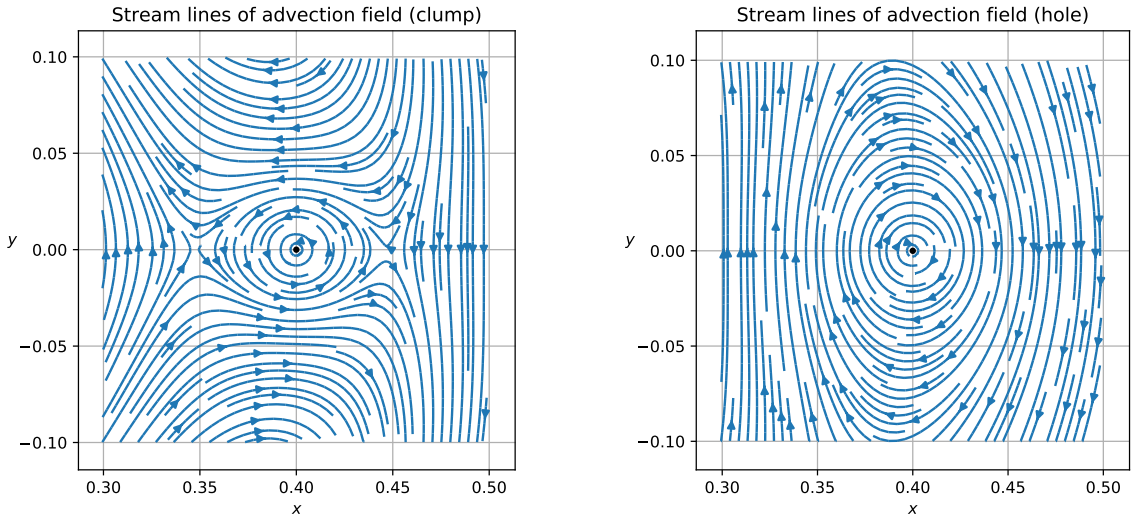

Fig. 12: Local stream lines of the advection field near positive (clump) and negative (hole) point-like vortices in a rotating frame.

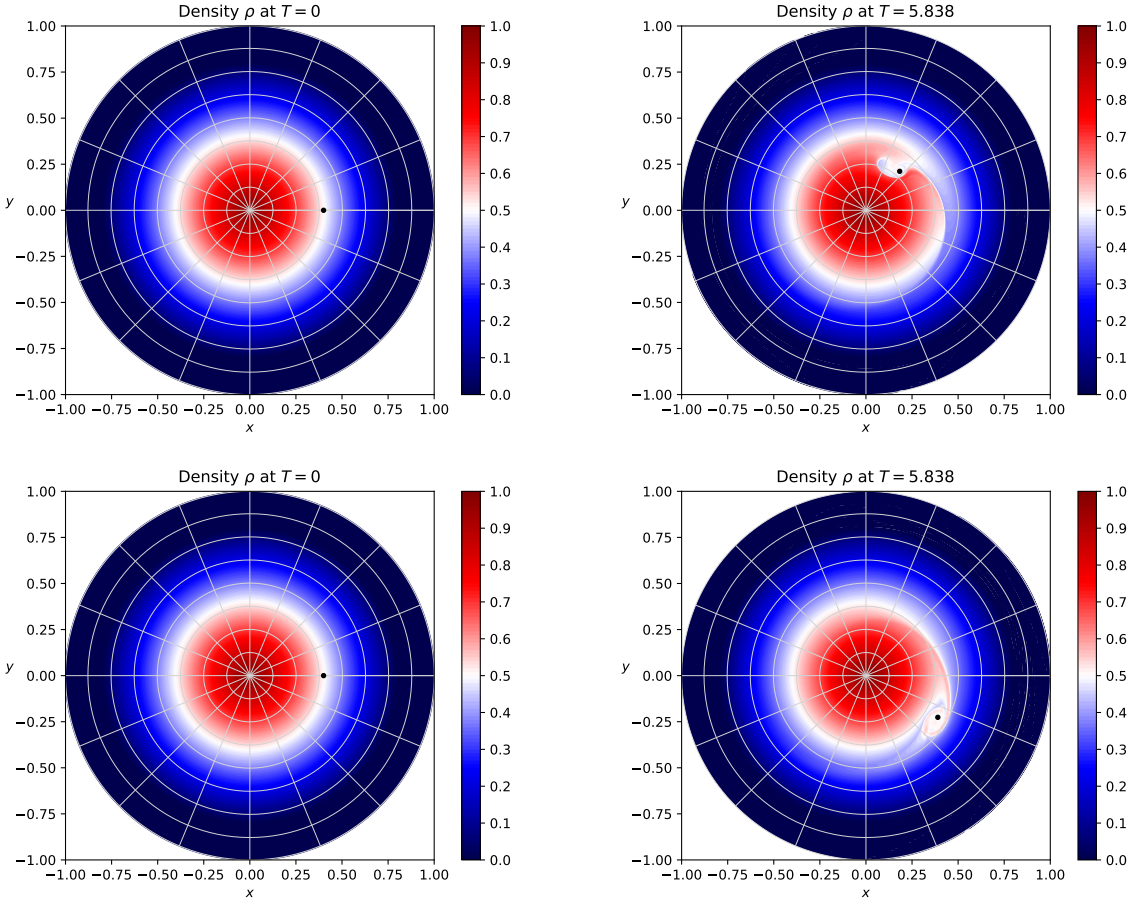

Fig. 13: Dynamics of positive (top) and negative (bottom) point-like vortices on a circular mapping: contour plot of the density at different times in a rotating frame.

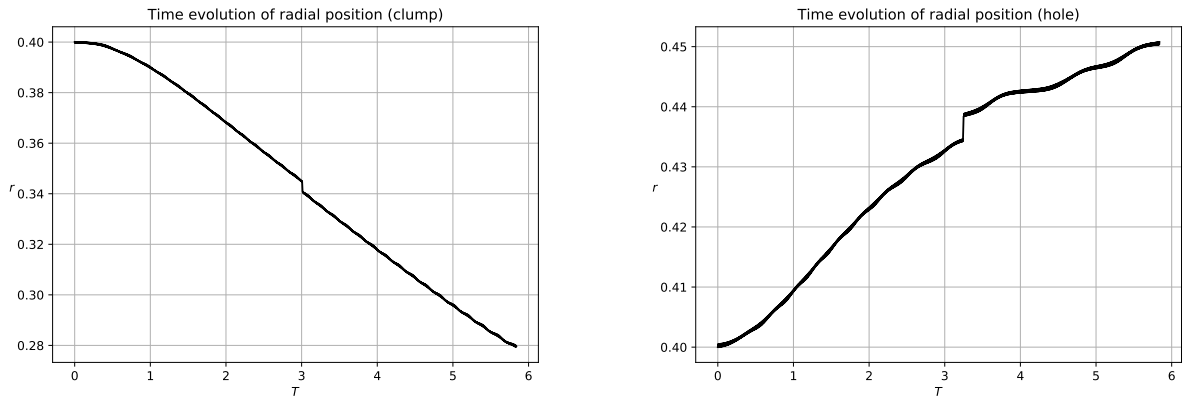

Fig. 14: Time evolution of the radial position of positive (clump) and negative (hole) point-like vortices.

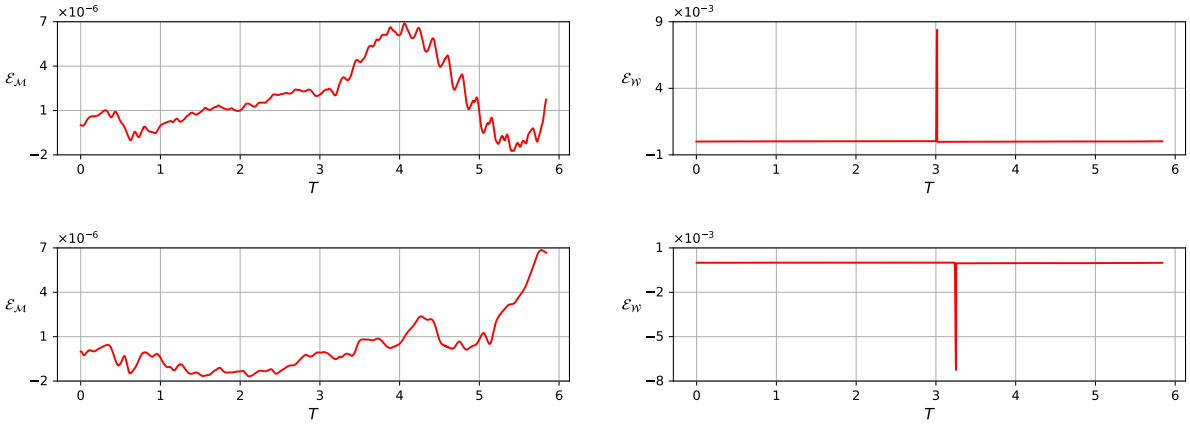

Fig. 15: Dynamics of positive (top) and negative (bottom) point-like vortices on a circular mapping: time evolution of the relative errors on the total mass (left) and energy (right).

procedure (21) with $f(\phi) = \phi^2$ and $\bar{\rho} = 1$, show the same qualitative behavior for clumps and holes (Figure 16). The final time $t = 35$ corresponds to the normalized time $T = 5.838$ considered before. For the conservation of mass and energy we get

$$\max_{t\in[0,35]} |\mathcal{E}_{\mathcal{M}}(t)| \approx 1.6 \times 10^{-5}\,, \qquad \max_{t\in[0,35]} |\mathcal{E}_{\mathcal{W}}(t)| \approx 6.9 \times 10^{-3}$$

for the positive point-like vortex and

$$\max_{t\in[0,35]} |\mathcal{E}_{\mathcal{M}}(t)| \approx 3.6 \times 10^{-6}\,, \qquad \max_{t\in[0,35]} |\mathcal{E}_{\mathcal{W}}(t)| \approx 5.7 \times 10^{-3}$$

for the negative point-like vortex. The time evolution of the relative errors on these conserved quantities is shown in Figure 17.

7. Conclusions and outlook

A comprehensive strategy for solving systems of coupled hyperbolic and elliptic partial differential equations as the 2D example (1) on mapped disk-like domains with a singularity at a unique pole, where one edge of the

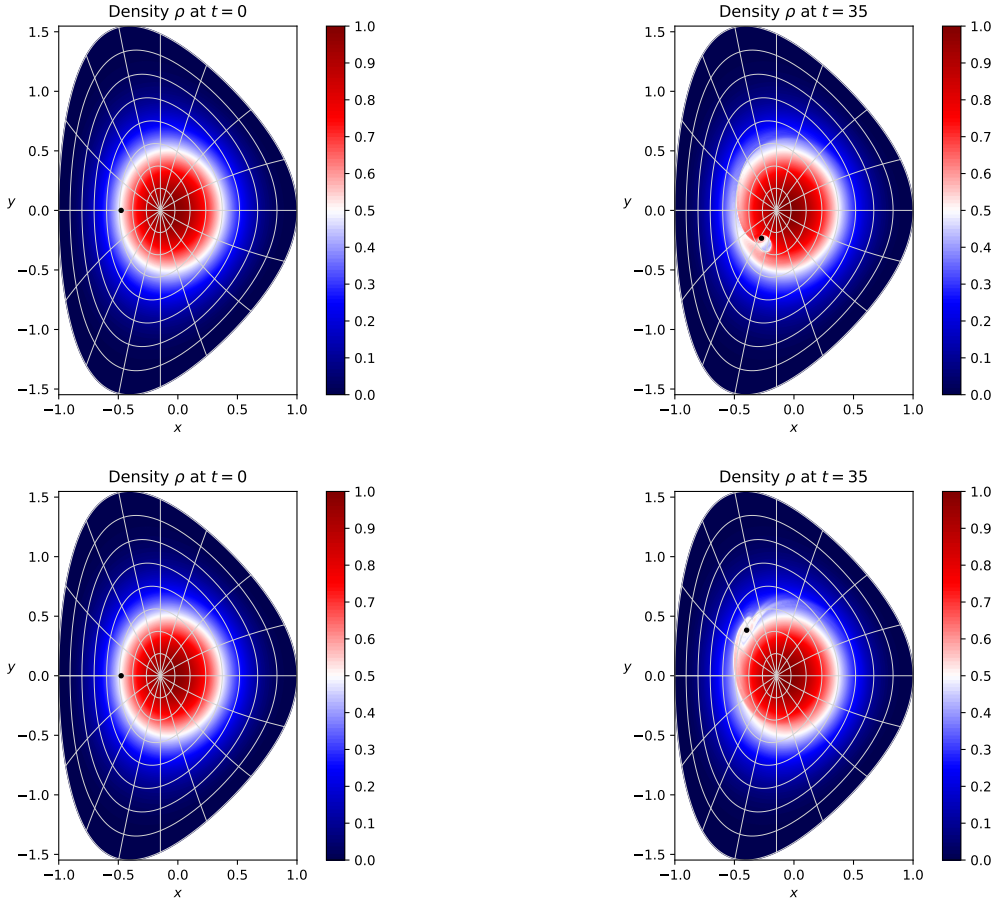

Fig. 16: Dynamics of positive (top) and negative (bottom) point-like vortices on the mapping (3) initialized with the parameters in Table 2: contour plot of the density at different times.

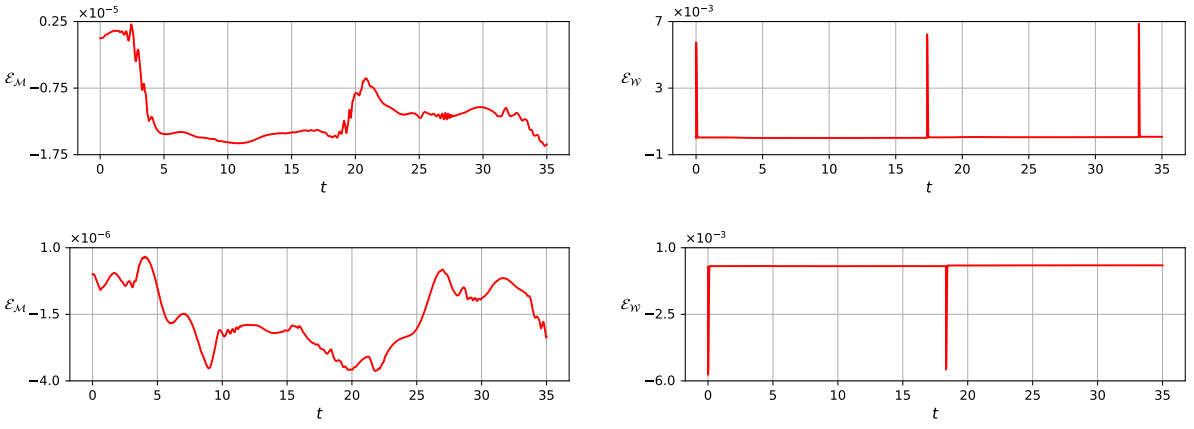

Fig. 17: Dynamics of positive (top) and negative (bottom) point-like vortices on the mapping (3) initialized with the parameters in Table 2: time evolution of the relative errors on the total mass (left) and energy (right).

rectangular logical domain is collapsed into one point of the physical domain, has been presented. A new set of coordinates, named pseudo-Cartesian coordinates, has been suggested for the time integration of the characteristics of

the hyperbolic equation of the system. Such coordinates turn out to be well-defined everywhere in the computational domain, including the pole, and provide a straightforward and relatively simple solution for dealing with singularities while solving advection problems. They reduce to standard Cartesian coordinates in the case of a circular mapping. Moreover, a finite element Poisson solver based on the recently developed arbitrarily smooth polar splines [1] has been implemented and tested. In the present work only C^1 smoothness was imposed at the pole, but a higher degree of smoothness, consistent with the spline degree, may be considered as well, if needed. Several test cases in the simplest case of a circular domain and in more complex geometries have been carried out; the numerical methods presented here show overall high-order convergence in the space discretization parameters, uniformly across the computational domain, including the pole. Moreover, the techniques discussed can be easily used in the context of particle-in-cell methods and are not necessarily restricted to semi-Lagrangian schemes, which were here discussed in more details. The range of physical problems that can be approached following the ideas presented in this work includes Vlasov-Poisson fully kinetic models as well as drift-kinetic and gyrokinetic models for magnetized plasmas, and turbulence models for incompressible inviscid Euler fluids in the context of fluid dynamics.

Acknowledgments

We would like to thank Eric Sonnendrücker for introducing us to the idea of using C^1 smooth spline basis functions and for constantly supporting this work, Ahmed Ratnani and Jalal Lakhlili for helping us with the implementation of the finite element Poisson solver and the choice of the data structure to be used for that purpose, Omar Maj and Camilla Bressan for helping us with the problem of finding numerical equilibria on complex mappings. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training program 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix A. Finite element Poisson solver: implementation

The code implementation is described using Fortran notation for multi-dimensional arrays, namely lower and upper bounds for a given index can be set arbitrarily (default lower bound, if not specified, is 1). As already pointed out, the linear system resulting from the finite element discretization of Poisson's equation is solved with an iterative conjugate gradient method. Therefore, we are only interested in discussing how we perform matrix-vector multiplications when storing matrices and vectors with the data structure discussed in the following. The stiffness and mass matrices A and M are stored in the stencil format

$$A(-p_1 : p_1, -p_2 : p_2, 1 : n_1, 1 : n_2)\,,$$

$$M(-p_1 : p_1, -p_2 : p_2, 1 : n_1, 1 : n_2)\,.$$

The arrays of coefficients u and f are stored in the stencil format

$$u(1 - p_1 : n_1 + p_1, 1 - p_2 : n_2 + p_2)\,,$$

$$f(1 - p_1 : n_1 + p_1, 1 - p_2 : n_2 + p_2)\,.$$

For given indices i_1, i_2, k_1 and k_2 we define

$$j_1^* = (i_1 + k_1 - 1) \mod (n_1) + 1\,,$$

$$j_2^* = (i_2 + k_2 - 1) \mod (n_2) + 1\,.$$

The matrix-vector multiplication is implemented as follows. Given two stencil vectors x and y and a stencil matrix S, we have

$$y(i_1, i_2) = \sum_{k_1=-p_1}^{p_1} \sum_{k_2=-p_2}^{p_2} S(k_1, k_2, i_1, i_2)\, x(j_1^*, j_2^*)\,,$$

with $i_1 = 1, \ldots, n_1$, $i_2 = 1, \ldots, n_2$. The buffer values are then filled as

$$y(1 - p_1 : 0, :) = y(1 - p_1 + n_1 : n_1, :)\,, \quad y(n_1 + 1 : n_1 + p_1, :) = y(1 : p_1, :)\,,$$

$$y(:, 1 - p_2 : 0) = y(:, 1 - p_2 + n_2 : n_2)\,, \quad y(:, n_2 + 1 : n_2 + p_2) = y(:, 1 : p_2)\,.$$

The C^1 projections of the stiffness and mass matrices are stored as block operators composed by the following four blocks:

$$\begin{pmatrix} B_{dd}(1:3, 1:3) & B_{sd}(1:3, 1:p_1, 1:n_2) \\ B_{ds}(1:p_1, 1:n_2, 1:3) & B_{ss}(-p_1:p_1, -p_2:p_2, 1:n_1-2, 1:n_2) \end{pmatrix}.$$

The matrix-vector multiplication for the off-diagonal blocks B_{sd} and B_{ds} is implemented as follows. Given a stencil vector x and a dense vector y, we have

$$y(i) = \sum_{j_1=1}^{p_1} \sum_{j_2=1}^{n_2} B_{sd}(i, j_1, j_2)\, x(j_1, j_2)\,,$$

with $i = 1, \ldots, 3$. Given a dense vector x and a stencil vector y, we have

$$y(i_1, i_2) = \sum_{j=1}^{3} B_{ds}(i_1, i_2, j)\, x(j)\,,$$

with $i_1 = 1, \ldots, p_1$ and $i_2 = 1, \ldots, n_2$. The buffer values are filled as described above and all remaining values are set to zero. The block L of the projection matrix containing the barycentric coordinates is also stored with three indices, namely $L(1:2, 1:n_2, 1:3)$ given by

$$L(i_1, i_2, l) = \lambda_l(c_{jk}^x, c_{jk}^y)\,,$$

with $i_1 = 1, 2$, $i_2 = 1, \ldots, n_2$, $l = 1, \ldots, 3$, $j = (i-1)/n_2 + 1$ and $k = (i-1) \mod (n_2) + 1$, with $i = (i_1 - 1)n_2 + i_2$. Here, $\lambda_l(c_{jk}^x, c_{jk}^y)$ are the barycentric coordinates evaluated at the spline coefficients c_{jk}^x, c_{jk}^y (control points) of the spline mapping. The C^1 projections of vectors are stored as block vectors composed by the following two blocks:

$$\begin{pmatrix} V_d(1:3) \\ V_s(1 - p_1 : (n_1 - 2) + p_1, 1 - p_2 : n_2 + p_2) \end{pmatrix}.$$

Given a stencil matrix S, the four blocks of its C^1 projection are obtained as follows. The block B_{dd} is given by

$$B_{dd}(i, j) = \sum_{i_1=1}^{2} \sum_{i_2=1}^{n_2} L(i_1, i_2, i)\, T(i_1, i_2, j)\,,$$

with $i, j = 1, \ldots, 3$ and T given by

$$T(i_1, i_2, l) = \sum_{k_1=-p_1}^{p_1} \sum_{k_2=-p_2}^{p_2} S(k_1, k_2, i_1, i_2)\, L(j_1^*, j_2^*, l)\,,$$

with $i_1 = 1, 2$, $i_2 = 1, \ldots, n_2$, $l = 1, \ldots, 3$, provided that $j_1^* \leq 2$ (all remaining elements are set to zero). The block B_{sd} is given by

$$B_{sd}(l, j_1^* - 2, j_2^*) = \sum_{k_1=-p_1}^{p_1} \sum_{k_2=-p_2}^{p_2} L(i_1, i_2, l)\, S(k_1, k_2, i_1, i_2)\,,$$

with $i_1 = 1, 2$, $i_2 = 1, \ldots, n_2$, $l = 1, \ldots, 3$, provided that $j_1^* \leq 2 + p_1$ (all remaining elements are set to zero). The block B_{ds} is given by

$$B_{ds}(i_1 - 2, i_2, l) = \sum_{k_1=-p_1}^{p_1} \sum_{k_2=-p_2}^{p_2} S(k_1, k_2, i_1, i_2)\, L(j_1^*, j_2^*, l)\,,$$

with $i_1 = 1, \ldots, 2 + p_1$, $i_2 = 1, \ldots, n_2$, $l = 1, \ldots, 3$, provided that $j_1^* \leq 2$ (all remaining elements are set to zero). The block B_{ss} is given by

$$B_{ss}(k_1, k_2, i_1, i_2) = S(k_1, k_2, i_1 + 2, i_2)\,,$$

with $i_1 = 1, \ldots, n_1 - 2$, $i_2 = 1, \ldots, n_2$, $k_1 = -p_1, \ldots, p_1$ and $k_2 = -p_2, \ldots, p_2$, provided that $j_1' = 2 + j_1$, with $j_1 = (i_1 - 1 + k_1) \mod (n_1 - 2)$ and $j_1' = (i_1 + 1 + k_1) \mod (n_1)$ (all remaining elements are set to zero). Given a stencil vector v, the two blocks of its C^1 projection are obtained as follows. The block V_d is given by

$$V_d(l) = \sum_{i_1=1}^{2} \sum_{i_2=1}^{n_2} L(i_1, i_2, l)\, v(i_1, i_2)\,,$$

with $l = 1, \ldots, 3$. The block V_s is given by

$$V_s(i_1 - 2, i_2) = v(i_1, i_2)\,,$$

with $i_1 = 3, \ldots, n_1$ and $i_2 = 1, \ldots, n_2$. The buffer values are set to zero along the s direction and filled applying periodicity along the θ direction. Homogeneous Dirichlet boundary conditions are imposed by setting

$$B_{ss}(k_1, k_2, i_1, i_2) = 0\,,$$

with $i_1 = 1, \ldots, n_1 - 2$, $i_2 = 1, \ldots, n_2$, $k_1 = -p_1, \ldots, p_1$ and $k_2 = -p_2, \ldots, p_2$, provided that $i > (n_1 - 3)n_2$ or $j > (n_1 - 3)n_2$, with $i = (i_1 - 1)n_2 + i_2$ and $j = (j_1 - 1)n_2 + j_2$, with $j_1 = (i_1 - 1 + k_1) \mod (n_1 - 2)$ and $j_2 = (i_2 - 1 + k_2) \mod (n_2)$, and by setting

$$V_s(n_1 - 2, i_2) = 0\,,$$

with $i_2 = 1, \ldots, n_2$.

References

[1] Deepesh Toshniwal, Hendrik Speleers, Ren R. Hiemstra, and Thomas J.R. Hughes. Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:1005–1061, 2017. doi: 10.1016/j.cma.2016.11.009.

[2] T. M. O'Neil. New Theory of Transport Due to Like-Particle Collisions. *Phys. Rev. Lett.*, 55:943–946, 1985. doi: 10.1103/PhysRevLett.55.943.

[3] D. H. E. Dubin and T. M. O'Neil. Two-dimensional guiding-center transport of a pure electron plasma. *Phys. Rev. Lett.*, 60:1286–1289, 1988. doi: 10.1103/PhysRevLett.60.1286.

[4] Ronald C Davidson. *Physics of Nonneutral Plasmas*. Imperial College Press and World Scientific Publishing Co. Pte. Ltd., 2001.

[5] C.F. Driscoll, D.Z. Jin, D.A. Schecter, and D.H.E. Dubin. Vortex dynamics of 2D electron plasmas. *Physica C: Superconductivity*, 369(1): 21–27, 2002. doi: 10.1016/S0921-4534(01)01216-3.

[6] M. Sengupta and R. Ganesh. Inertia driven radial breathing and nonlinear relaxation in cylindrically confined pure electron plasma. *Physics of Plasmas*, 21(2), 2014. doi: 10.1063/1.4866022.

[7] M. Sengupta and R. Ganesh. Linear and nonlinear evolution of the ion resonance instability in cylindrical traps: A numerical study. *Physics of Plasmas*, 22(7), 2015. doi: 10.1063/1.4927126.

[8] R. Ganesh and J. K. Lee. Formation of quasistationary vortex and transient hole patterns through vortex merger. *Physics of Plasmas*, 9(11): 4551–4559, 2002. doi: 10.1063/1.1513154.

[9] David A. Schecter and Daniel H. E. Dubin. Vortex Motion Driven by a Background Vorticity Gradient. *Phys. Rev. Lett.*, 83:2191–2194, 1999. doi: 10.1103/PhysRevLett.83.2191.

[10] D. A. Schecter, D. H. E. Dubin, K. S. Fine, and C. F. Driscoll. Vortex crystals from 2D Euler flow: Experiment and simulation. *Physics of Fluids*, 11(4):905–914, 1999. doi: 10.1063/1.869961.

[11] David A. Schecter and Daniel H. E. Dubin. Theory and simulations of two-dimensional vortex motion driven by a background vorticity gradient. *Physics of Fluids*, 13(6):1704–1723, 2001. doi: 10.1063/1.1359763.

[12] L.A. Artsimovich. Tokamak devices. *Nuclear Fusion*, 12(2):215, 1972.

[13] J. S. Sawyer. A semi-Lagrangian method of solving the vorticity advection equation. *Tellus*, 15(4):336–342, 1963. doi: 10.3402/tellusa.v15i4.8862.

[14] Eric Sonnendrücker, Jean Roche, Pierre Bertrand, and Alain Ghizzo. The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation. *Journal of Computational Physics*, 149(2):201–220, 1999. doi: 10.1006/jcph.1998.6148.

[15] N. Besse and E. Sonnendrcker. Semi-Lagrangian schemes for the Vlasov equation on an unstructured mesh of phase space. *Journal of Computational Physics*, 191(2):341–376, 2003. doi: https://doi.org/10.1016/S0021-9991(03)00318-8.

[16] V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrücker, J. Vaclavik, and L. Villard. A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation. *Journal of Computational Physics*, 217(2):395–423, 2006. doi: https://doi.org/10.1016/j.jcp.2006.01.023.

[17] V. Grandgirard, Y. Sarazin, X. Garbet, G. DifPradalier, Ph. Ghendrih, N. Crouseilles, G. Latu, E. Sonnendrücker, N. Besse, and P. Bertrand. GYSELA, a full-f global gyrokinetic Semi-Lagrangian code for ITG turbulence simulations. *AIP Conference Proceedings*, 871(1):100–111, 2006. doi: 10.1063/1.2404543.

[18] Nicolas Crouseilles, Michel Mehrenberger, and Eric Sonnendrücker. Conservative semi-Lagrangian schemes for Vlasov equations. *Journal of Computational Physics*, 229(6):1927–1953, 2010. doi: 10.1016/j.jcp.2009.11.007.

[19] V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, Ch. Ehrlacher, D. Esteve, X. Garbet, Ph. Ghendrih, G. Latu, M. Mehrenberger, C. Norscini, Ch. Passeron, F. Rozar, Y. Sarazin, E. Sonnendrcker, A. Strugarek, and D. Zarzoso. A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations. *Computer Physics Communications*, 207:35–68, 2016. doi: https://doi.org/10.1016/j.cpc.2016.05.007.

[20] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928. doi: 10.1007/BF01448839.

[21] N. Bouzat, C. Bressan, V. Grandgirard, G. Latu, and M. Mehrenberger. Targeting realistic geometry in Tokamak code Gysela. *ArXiv e-prints*, 2017.

[22] Olivier Czarny and Guido Huysmans. Bézier surfaces and finite elements for MHD simulations. *Journal of Computational Physics*, 227(16): 7423–7445, 2008. doi: 10.1016/j.jcp.2008.04.001.

[23] William J. Gordon and Richard Riesenfeld. *B-spline curves and surfaces*. In R. E. Barnhill and R. F. Riesenfeld, editors, Computer Aided Geometric Design, Academic Press, Inc., 1974.

[24] Gerald Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, 1993.

[25] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

[26] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Springer Science & Business Media, 2010.

[27] Tao Xiong, Giovanni Russo, and Jing-Mei Qiu. High Order Multi-dimensional Characteristics Tracing for the Incompressible Euler Equation and the Guiding-Center Vlasov Equation. *Journal of Scientific Computing*, 77(1):263–282, 2018. doi: 10.1007/s10915-018-0705-y.

[28] Tatsuoki Takeda and Shinji Tokuda. Computation of MHD Equilibrium of Tokamak Plasma. *Journal of Computational Physics*, 93(1):1–107, 1991. doi: 10.1016/0021-9991(91)90074-U.

[29] M. T. Bettencourt. Controlling Self-Force for Unstructured Particle-in-Cell (PIC) Codes. *IEEE Transactions on Plasma Science*, 42(5): 1189–1194, 2014. doi: 10.1109/TPS.2014.2313515.