

Discontinuous Galerkin methods for plasma physics in the scrape-off layer of tokamaks

C. Michoski[‡]*, D. Meyerson[‡], T. Isaac^{*}, F. Waelbroeck[‡],
Institute for Fusion Studies (IFS)[‡], Department of Physics[‡]
Institute for Computational Engineering and Sciences (ICES)^{},*
Department of Aerospace Engineering and Engineering Mechanics[‡]
University of Texas at Austin, Austin, TX 78712

Abstract

A new parallel discontinuous Galerkin solver, called ArcOn, is developed to describe the intermittent turbulent transport of filamentary blobs in the scrape-off layer (SOL) of fusion plasma. The model is comprised of an elliptic subsystem coupled to two convection-dominated reaction-diffusion-convection equations. Upwinding is used for a class of numerical fluxes developed to accommodate cross product driven convection, and the elliptic solver uses SIPG, NIPG, IIPG, Brezzi, and Bassi-Rebay fluxes to formulate the stiffness matrix. A novel entropy sensor is developed for this system, designed for a space-time varying artificial diffusion/viscosity regularization algorithm. Some numerical experiments are performed to show convergence order on manufactured solutions, regularization of blob/streamer dynamics in the SOL given unstable parameterizations, long-time stability of modon (or dipole drift vortex) solutions arising in simulations of drift-wave turbulence, and finally the formation of edge mode turbulence in the scrape-off layer under turbulent saturation conditions.

Keywords: Nodal/modal, discontinuous Galerkin, unified formulation, elliptic, hyperbolic, parabolic, plasma physics, fusion, tokamaks, verification, blobs, modons, scrape-off layer, interior penalty, turbulence, upwinding, artificial diffusion, massively parallel.

Contents

§1	Introduction	2
§2	The governing equations	4
§3	The discretization procedure	5
3.1	The approximation procedure	6
3.1.1	The continuity equation	6
3.1.2	The ion polarization	7
3.1.3	The convective fluxes	7
3.1.4	The elliptic subsystems	9
3.2	The formulation	11

[‡]Corresponding author, [†]michoski@ices.utexas.edu

§4 Numerical tests	12
4.1 Elliptic verification	13
4.2 Hyperbolic-parabolic verification	13
4.3 Streamer validation	16
§5 Regularization processes	18
5.1 Artificial diffusion and gradient driven flow	18
§6 Physical solutions	22
6.1 Modon solutions	22
6.2 Turbulence	25
§7 Conclusion	27
§8 Acknowledgements	28

§1 Introduction

In magnetic fusion devices, a large fraction of the thermal power flows to an actively cooled strike-plate through a narrow layer called the scrape-off layer (SOL) [44]. This layer lies at the boundary between closed field lines and field lines that connect to the wall, and its width is determined by the competition between the turbulent transport across the magnetic field and the very rapid transport along the field. This width determines the heat flux density on the wall and impacts wall erosion, recycling, and density control. Transport in the SOL is known to be highly intermittent and dominated by the ejection of coherent plasma filaments known as “blobs” or “streamers” [22, 35, 64]. The modeling of this turbulent transport is important in order to predict and understand the dependence of the SOL width on the plasma parameters and machine size.

The equations that govern SOL blob dynamics have been formulated and analyzed [5, 26, 35, 62], but many questions remain open. From a mathematical point of view the dynamics of existing SOL plasma models can be characterized as systems of equations that are driven by operators that are nonlinear and contain multiple signatures (e.g. elliptic, parabolic, dispersive, hyperbolic operators, etc.) displaying weak regularity features in the coupling. As the consistency, stability, and accuracy of numeric methods are strongly constrained by the behavior of the underlying mathematical models, the difficulties posed by these complicated SOL plasma dynamics raise interesting and challenging technical questions.

As a potential solution to the difficulties that can arise, we explore the use of a discontinuous Galerkin (DG) finite element method for modeling blob dynamics in the SOL. The discontinuous Galerkin method is a high order numerical method, that has been found to provide formidable accuracy, stability, and robustness in many areas of nonlinear dynamics [15, 16, 25, 37]. Moreover, DG methods are unified in the sense that they are well-suited for rigorous analysis of both the physics as well as the numerics and mathematics of the system. They demonstrate high order convergence rates [2, 4], are well-established as candidates for computationally optimal adaptive technologies (e.g. *hp*-adaptivity and *r*-adaptivity) [21, 52], are extremely scalable (especially on state-of-the-art architectures with high thread parallel arithmetic intensity, such as GPUs and coprocessors) [1, 34, 73, 75], and are often noted as being remarkably flexible for accurately modeling large categories of coupled systems of

initial-boundary value problems with strongly nonlinear forcings. This aspect of DG methods makes them particularly appealing for studying scrape-off layer dynamics, as the system of PDEs in question in the scrape-off layer is often highly variable, requires great flexibility in representation, possess weak regularity features, and may involve complicated geometries (e.g. in the presence of magnetic chaos) with nonlinear boundary forcings that are befitting to finite element methods in general.

The application of DG methods to problems arising in plasma physics and nuclear engineering have established a high benchmark. Dawson, Wheeler and Proft studied neutron transport in [20]. In [31, 32] Hesthaven and Jacobs addressed DG plasma building block models, along with providing high resolution insight into the basic aspects of applying DG methods to plasma problems. Warburton and Karniadakis developed a turbulent DNS solver for unsteady viscous MHD in [72], and Cheng, Li, Qiu, and Xu [14] have recently studied specialized positivity preserving DG methods in the context of ideal MHD. In plasma kinetic theory, Heath, Gamba, Morrison, Cheng, and Michler [13, 28] have developed some impressive high dimensional Vlasov-Poisson schemes, where Rossmanith and Seal have similarly implemented DG schemes for Vlasov-Poisson, though in the Semi-Lagrangian setting [61]. Edge turbulence models have recently worked their way into some studies of Peterson and Hammet[59], while some of the most well-established DG results in computational plasma physics are those from the multifluid models of Shumlak, Loverich, Hakim, Srinivasan, and Meier et al. [45, 46, 48, 67, 68], which have not only been thoroughly validated, but thoroughly benchmarked against established codes in the community.

The present paper represents an interdisciplinary effort to apply general discontinuous Galerkin methods to the modeling of plasma dynamics in tokamak scrape-off layers. This work draws heavily from a number of disciplines, in the context of trying to serve as a powerful staging ground for further and deeper analysis of tokamak reactors. In this direction, the code architecture we have developed for this study has been given the moniker ArcOn. The code itself is explicit in time and consists of three coupled equations that require two local discontinuous solves, and a global linear solve in the elliptic subsystem (to be described below). It has been written (primarily) in C++, and utilizes a number of external libraries, including: deal.II [7], p4est [12], and PETSc [6]. The development of ArcOn has required the modification of deal.II, which required a fundamental update/extension to core functionality in order to provide inter-software support for periodic boundary data over massively parallel distributed meshes. We mention this process briefly here, as it was quite time-consuming and introduced a nontrivial technical challenge in the computer science aspect of the code development. These modifications were performed in consultation with project leaders, and have subsequently been openly distributed to their respective communities for broad use.

All computations below were run in parallel on the Texas Advanced Computing Center's (TACC's) 10 PetaFLOPS Dell Linux Cluster based Stampede system, comprised of 6,400+ Dell PowerEdge server nodes, each outfitted with 2 Intel Xeon E5 (Sandy Bridge) processors and an Intel Xeon Phi Coprocessor (MIC Architecture). For this study, relatively small meshes were used at low polynomial order, that ranged from tens of thousands of degrees of freedom, to tens of millions. All runs were performed in parallel on between 16 to 256 cores.

An outline of the paper is as follows. In section 2 we present the physics-based SOL model for filamentary blob transport. Here we provide motivation for the system formulation, and heuristically discuss its features, scope, and limitations. In section 3 we review the formulation of the discontinuous Galerkin method applied to the mathematical model presented in section 2. We address some salient features of ArcOn, including the ability to use both multiple solvers (e.g. SIPG, NIPG, etc.) and various basis functions, including both nodal and modal finite elements. We also describe the upwind-

ing flux schemes used in the approximate Riemann-type solvers, and the strong stability preserving temporal discretization scheme employed. In section 4 we discuss the parameters of a classical verification study by way of a manufactured solution, and perform numerical experiments that demonstrate the optimal (to super-optimal in places) convergence order in each operator subsystem of ArcOn. In section 5 we discuss some of the basic stability features of the DG method, and how regularization procedures can be used to ensure numerical stability. We present here a basic blob example adjoined to a new artificial diffusion scheme, and run using a CFL-based variable timestepping algorithm. Finally in section 6 we show a modon solution to the physics of the system, briefly analyze and validate some of the cogent numerical properties of blobs, and demonstrate the effectiveness of ArcOn in solving turbulent plasma dynamics at saturation..

§2 The governing equations

We use a two-dimensional model for the SOL turbulence that describes the evolution of the density n , the vorticity U , and the electrostatic potential ϕ . The latter plays the role of a stream-function for the ion velocity, which is given by the perpendicular $\mathbf{E} \times \mathbf{B}$ drift velocity $V_{\mathbf{E} \times \mathbf{B}} \sim \mathbf{b} \times \nabla \phi$ (here $\mathbf{b} = \mathbf{B}/B$ and $\mathbf{E} = -\nabla \phi$ is the electric field). The \mathbf{e}_z direction is chosen to lie along the magnetic field, $\mathbf{e}_z = \mathbf{b}$. The model is based on the assumption that due to rapid transport along the magnetic field, the quantities of interest have negligible variation in the z direction except at the boundary between the plasma and the wall where sheath physics enters the equations as a sink term. The coordinate along the normal to the plasma surface is denoted by x , and the coordinate in the binormal direction (perpendicular to both \mathbf{e}_x and the magnetic field) is denoted by y . The model equations are solved using the variables $(x, y, t) \in \Omega \times (0, T)$, where Ω is a domain of the (x, y) plane extending over the SOL in the x direction. The spatial units are normalized relative to a ratio of the ion acoustic speed c_s and the ion gyrofrequency ω_{ci} , given in terms of the Bohm gyro-radius ρ_{cs} . The equations are

$$\partial_t n + [\phi, n] = -\alpha_0 n + D \nabla_{\perp}^2 n, \quad n|_{t=0} = n_0, \quad n = n_b \text{ on } \partial\Omega, \quad (2.1)$$

$$\partial_t U + [\phi, U] = \alpha \phi - \beta [x, \ln n] + \mu \nabla_{\perp}^2 U, \quad U|_{t=0} = U_0, \quad U = U_b \text{ on } \partial\Omega, \quad (2.2)$$

$$U = \nabla_{\perp}^2 \phi, \quad \phi|_{t=0} = \phi_0, \quad \phi = \phi_b \text{ on } \partial\Omega, \quad (2.3)$$

corresponding to the continuity, vorticity and Poisson equations respectively. The brackets are defined for scalar-valued functions a and b by, $[a, b] = \mathbf{e}_z \cdot (\nabla_{\perp} a \times \nabla_{\perp} b)$. The left-hand sides of (2.1) and (2.2) represents the convective derivative with respect to the ion fluid. We will also use $\nabla_{\perp} \cdot \nabla_{\perp} = \nabla_{\perp}^2 = \Delta_{\perp}$ frequently throughout to represent orthogonal operators. In addition, we will frequently represent our coordinate chart by pairs in vector form, $\mathbf{x} = (x, y)$.

The first equation (2.1) indicates mass conservation of the plasma density $n = n(\mathbf{x}, t)$. The plasma (mass) diffusion on the right is driven by the diffusion coefficient D . The reaction-type term $\alpha_0 n$ represents the competition between ionization sourcing and parallel loss at the wall, with $\alpha_0 \sim (k\alpha - L_{\parallel}/c_s)$, where L_{\parallel} is the parallel length from the mid-plane to the wall and c_s the ion-acoustic speed.

The second equation (2.2) is frequently referred to as a conservation equation for the vorticity $U = U(\mathbf{x}, t)$. Note that the vorticity variable U is introduced for the sake of clarity, so as to separate the elliptic equation (2.3) and the dynamic equation (2.2) into two well-defined mathematical subsystems that will be treated individually in our numerical methods. The vorticity, up to a rescaling, is

determined by the electric field potential $\phi = \phi(\mathbf{x}, t)$, leading to the Boussinesq approximation in the ion polarization term,

$$\nabla \cdot \frac{D}{Dt} \left(\frac{nMc^2}{B^2} \nabla_{\perp} \phi \right) \propto \partial_t U + [\phi, U], \quad (2.4)$$

where M is the ion mass, and c is the speed of light. The term proportional to β represents the effects of ∇B (caused by toroidal curvature) on the system dynamics. Lastly, the term proportional to μ can be viewed as a manifestation of classical viscosity, though in practice its magnitude is generally chosen so that it serves as a numerical regularization term [26, 35].

Below we will occasionally make the implicit assumption that β is a linear function of n , $\beta = Cn$ for C a constant, such that the leading n^{-1} term can be dropped, leading to a stabilized form about the vacuum, $n \approx 0$ (though this instability can always be removed through a basic rescaling argument, as presented in section 6).

§3 The discretization procedure

Let us discretize our spatial domain Ω . Consider the open set $\Omega \subset \mathbb{R}^2$ with physical boundary $\partial\Omega$, given $T > 0$ such that $\mathcal{Q}_T = \Omega \times (0, T)$. Let \mathcal{T}_h denote the partition of the closure of the polygonal mesh of Ω , which we denote Ω_h , into a finite number of polygonal elements denoted Ω_e , such that $\mathcal{T}_h = \{\Omega_{e_1}, \Omega_{e_2}, \dots, \Omega_{e_{n_e}}\}$, for $n_e \in \mathbb{N}$ the number of elements in Ω_h . Here and below the mesh diameter h is chosen to satisfy $h = \min_{ij}(d_{ij})$ for the distance function $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ and elementwise face vertices $\mathbf{x}_i, \mathbf{x}_j \in \partial\Omega_e$ when the mesh is structured and regular. For unstructured or irregular meshes we provide a range over the mesh.

Now, let Γ_{ij} denote the face shared by two neighboring elements Ω_{e_i} and Ω_{e_j} , and for $i \in I \subset \mathbb{Z}^+ = \{1, 2, \dots\}$ define the indexing set $r(i) = \{j \in I : \Omega_{e_j} \text{ is a neighbor of } \Omega_{e_i}\}$. Let us denote all Ω_{e_i} containing the boundary $\partial\Omega_h$ by S_j and letting $I_B \subset \mathbb{Z}^- = \{-1, -2, \dots\}$ define $s(i) = \{j \in I_B : S_j \text{ is a face of } \Omega_{e_i}\}$ such that $\Gamma_{ij} = S_j$ for $\Omega_{e_i} \in \Omega_h$ when $S_j \in \partial\Omega_{e_i}$, $j \in I_B$. Then for $\Xi_i = r(i) \cup s(i)$, we have

$$\partial\Omega_{e_i} = \bigcup_{j \in \Xi(i)} \Gamma_{ij}, \quad \text{and} \quad \partial\Omega_{e_i} \cap \partial\Omega_h = \bigcup_{j \in s(i)} \Gamma_{ij}.$$

We are interested in obtaining an approximate solution to (2.1)-(2.3) at time t on the finite dimensional space of piecewise polynomial functions over Ω restricted to \mathcal{T}_h , given as

$$S_h^p(\Omega_h, \mathcal{T}_h) = \{v : v|_{\Omega_{e_i}} \in \mathcal{P}^p(\Omega_{e_i}) \quad \forall \Omega_{e_i} \in \mathcal{T}_h\}$$

for $\mathcal{P}^p(\Omega_{e_i})$ the space of degree of (at most) p polynomials over Ω_{e_i} .

Choosing a set of basis functions N_l for $l = 0, \dots, n_p$ the corresponding degrees of freedom indexed by the (spatial) spectral order p of the local element, we can denote the three local solutions in the *modal basis* over a finite element cell Ω_{e_i} as

$$n_{hp}^i(t, \mathbf{x}) = \sum_{l=0}^{n_p} n_l^i(t) N_l^i(\mathbf{x}), \quad \text{and} \quad U_{hp}^i(t, \mathbf{x}) = \sum_{l=0}^{n_p} U_l^i(t) N_l^i(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_{e_i},$$

$$\phi_{hp}^i(t, \mathbf{x}) = \sum_{l=0}^{n_p} \phi_l^i(t) N_l^i(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_{e_i},$$

where the N_l^i 's are the finite element shape functions. Let us consider j arbitrary finite dimensional test functions ζ_j characterized by

$$\zeta_j^i(\mathbf{x}) = \sum_{l=0}^{n_p} \zeta_l^i N_l^i(\mathbf{x}), \quad \forall (\mathbf{x}) \in \Omega_{e_i},$$

where ζ_j^i are the modal coordinates in each Ω_{e_i} . These functions are chosen as members of the broken Sobolev space over the partition \mathcal{T}_h defined by

$$W^{k,q}(\Omega_h, \mathcal{T}_h) = \{\omega : \omega|_{\Omega_{e_i}} \in W^{k,q}(\Omega_{e_i}) \quad \forall \Omega_{e_i} \in \mathcal{T}_h\}.$$

In particular, when $q = 2$ and $k = 0$ we have the standard L^2 -norm, such that $\zeta_j \in L_{loc}^2$ with piecewise polynomials representations $\zeta_j^i \in S_h^p(\Omega_h, \mathcal{T}_h)$. For the d -dimensional vector basis (in our case the spatial dimension $\dim = d = 2$) we will denote the j arbitrary finite dimensional test functions ξ_j characterized as the direct product space, such that in the usual sense, when $\zeta_j \in L^2$ with piecewise polynomials $\zeta_j^i \in S_h^p(\Omega_h, \mathcal{T}_h)$, this implies that $\xi_j \in [L^2]^2$ with piecewise polynomials $\xi_j^i \in [S_h^p(\Omega_h, \mathcal{T}_h)]^2$, and so on.

We can equivalently recast the modal discontinuous Galerkin solution into the *nodal basis* (i.e. the nodal discontinuous Galerkin solution), by choosing \tilde{n}_p support points in each finite element cell,

$$\mathbf{V}_{hp}^i(t, \mathbf{x}) = \sum_{l=0}^{n_p} \mathbf{V}_l^i(t) \zeta_l^i(\mathbf{x}) = \sum_{\ell=0}^{\tilde{n}_p} \mathbf{V}_\ell^i(\mathbf{x}_\ell^i, t) \varphi_\ell^i(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_{e_i},$$

for nodal basis φ_ℓ^i . It should be observed that the nodal and modal solutions are not equivalent, as discussed in detail in [30]. We also note that ArcOn at present supports 19 different basis functions, including discontinuous tensor products of Legendre polynomials, Lagrange nodal polynomials, etc.

3.1 The approximation procedure

Next we recast (2.1)–(2.3) in the auxiliary mixed formulation, as satisfying the six simultaneous equations

$$\partial_t n + [\phi, n] = -\alpha n + D \nabla_\perp \cdot \mathcal{O}, \quad \mathcal{O} = \nabla_\perp n, \quad (3.1)$$

$$\partial_t U + [\phi, U] = \alpha \phi - \beta[x, \ln n] + \mu \nabla_\perp \cdot \mathcal{U}, \quad \mathcal{U} = \nabla_\perp U, \quad (3.2)$$

$$\nabla_\perp \cdot \mathbf{E}_\perp = U, \quad \mathbf{E}_\perp = \nabla_\perp \phi, \quad (3.3)$$

where we note that substitutions can be made in terms of the ‘‘auxiliary variables’’ $\{\mathcal{O}, \mathcal{U}, \mathbf{E}_\perp\}$.

3.1.1 The continuity equation

In order to recover the weak (or variational) form for the finite element solution, we multiply each equation by a different finite element test function ζ and integrate by parts. Thus the coupled equations of (6.2) become the pair

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_{e_i}} n \zeta_1^i dx dy &= \int_{\partial \Omega_{e_i}} D \mathcal{O} \zeta_1^i \cdot \mathbf{n} dx dy - \int_{\Omega_{e_i}} \zeta_1^i (\nabla_\perp \phi \times \nabla_\perp n) dx dy \\ &\quad - \int_{\Omega_{e_i}} \alpha n \zeta_1^i dx dy - \int_{\Omega_{e_i}} D \mathcal{O} \nabla_\perp \zeta_1^i dx dy, \\ \int_{\Omega_{e_i}} \mathcal{O} \cdot \xi_1^i dx dy &= \int_{\partial \Omega_{e_i}} n \xi_1^i \cdot \mathbf{n} dx dy - \int_{\Omega_{e_i}} n \nabla_\perp \cdot \xi_1^i dx dy. \end{aligned} \quad (3.4)$$

This variational formulation is then projected into the discrete discontinuous space. The mass diffusion terms on the RHS of (3.4) are solved using the unified mixed method form for elliptic and parabolic solutions to PDEs [2, 4, 60], where we denote the boundary term by $\mathfrak{D}_i(\mathcal{O}_{hp}^i, n_{hp}^i, \zeta_1^i)$. In this generalized setting we are able to select from a large collection of diffusive numerical fluxes, such as the local discontinuous Galerkin flux (LDG), the interior penalty Galerkin flux (IPG), the symmetric interior penalty Galerkin flux (SIPG), and the nonsymmetric interior penalty Galerkin flux (NIPG), etc. Effectively this reduces to selecting a particular type of numerical behavior that relates to stability, consistency, and convergence properties of the numerical solution, which will be discussed in more detail below. The auxiliary equation in (3.4) is formatted in essentially the same way, where here the boundary flux is represented by $\mathcal{N}_i(n_{hp}^i, \xi_1^i)$.

3.1.2 The ion polarization

The variational form of the vorticity equation (6.3) is almost the same, but with the addition of the curvature term:

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_{e_i}} \zeta_2^i U dx dy &= - \int_{\Omega_{e_i}} \zeta_2^i (\nabla_{\perp} \phi \times \nabla_{\perp} U) dx dy - \int_{\Omega_{e_i}} \zeta_2^i \beta (\nabla_{\perp} x \times n^{-1} \nabla_{\perp} n) dx dy \\ &\quad + \int_{\Omega_{e_i}} \zeta_2^i \alpha \phi dx dy + \int_{\partial\Omega_{e_i}} \zeta_2^i \mu \mathcal{U} \cdot \mathbf{n} dx dy - \int_{\Omega_{e_i}} \mu \mathcal{U} \nabla_{\perp} \zeta_2^i dx dy, \\ \int_{\Omega_{e_i}} \xi_2^i \cdot \mathcal{U} dx dy &= \int_{\partial\Omega_{e_i}} \xi_2^i \cdot U \mathbf{n} dx dy - \int_{\Omega_{e_i}} U \nabla_{\perp} \cdot \xi_2^i dx dy, \end{aligned} \quad (3.5)$$

where again the cross products in the ion polarization term represent the same choices as those given above.

As above with the continuity equation, we now project the weak formulation into the piecewise polynomial representation space. The diffusive terms of (3.5) are solved in the same way as those in (3.4), where $\mathcal{M}_i(\mathcal{U}_{hp}^i, \phi_{hp}^i, \zeta_2^i)$ represents the diffusive boundary term, and the auxiliary numerical boundary term is given by $\hat{\mathcal{L}}_i$.

3.1.3 The convective fluxes

The convective fluxes for this system demonstrate some unique difficulties in comparison to those presented by standard divergence form flux-based solvers. Namely, the bracketed term is a cross product of gradients that has no exact algebraic divergence form Jacobian. That is, even though for example $[a, b] = \nabla \cdot (a \mathbf{e}_z \times \nabla b)$, the arguments to the divergence operator are differential and not algebraic. Hence the usual Jacobian matrix of the term is different than those often encountered in computational fluid dynamics, where standard implementations of Riemann solvers such as Roe or Lax-Friedrichs are straight forward to implement. These bracketed fluxes are however, quite common in plasma physics, and thus must be addressed. We do so here by applying a standard upwinding approach to each equation separately, thus inducing a slightly weaker coupling than the some common Riemann solvers (e.g. the Roe solver). It remains an open question what flux formulation might be used to strengthen this coupling, but in this direction we point the reader to the potential of using cosymplectic forms [56] to tensorialize the flux Jacobian into an eigenproblem.

To form our upwinding scheme, we first notice that expanding the cross product term of the mass equation generates twisted mixed partials,

$$- \int_{\Omega_{e_i}} \zeta_1^i (\nabla_{\perp} \phi \times \nabla_{\perp} n) dx dy = - \int_{\Omega_{e_i}} \zeta_1^i \frac{\partial \phi}{\partial x} \frac{\partial n}{\partial y} dx dy + \int_{\Omega_{e_i}} \zeta_1^i \frac{\partial n}{\partial x} \frac{\partial \phi}{\partial y} dx dy. \quad (3.6)$$

that drives the convective transport of the density.

A naive coupling can be chosen that simply linearizes the solution relative to the auxiliary terms, corresponding to an electric field $\mathbf{E}_{\perp} = \nabla_{\perp} \phi = (E_x, E_y)$ and density gradient $\mathcal{O} = (\mathcal{O}_x, \mathcal{O}_y)$, such that (3.6) can be written as

$$\underline{\text{The quasistrong flux}} : \left\{ \int_{\Omega_{e_i}} \zeta_1^i (\nabla_{\perp} \phi \times \nabla_{\perp} n) dx dy = \int_{\Omega_{e_i}} \zeta_1^i (\mathbf{E}_{\perp} \times \mathcal{O}) dx dy. \quad (3.7) \right.$$

This form of (3.6) is characterized as a quasistrong linearization as it forces the convective transport through the diffusive fluxes, as can be seen more explicitly below.

An alternative to (3.7) is to expand (3.6) into a stronger form, by rebalancing the components of (3.6) across the boundary terms, thus inducing a proper convective flux as:

$$\underline{\text{The strong flux}} : \left\{ \begin{aligned} - \int_{\Omega_{e_i}} \zeta_2 \frac{\partial \phi}{\partial x} \frac{\partial n}{\partial y} dx dy &= -\frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 \phi \mathcal{O}_y n_x dx dy + \frac{1}{2} \int_{\Omega_{e_i}} \phi \mathcal{O}_y \frac{\partial \zeta_2}{\partial x} dx dy \\ &\quad - \frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 n E_x n_y dx dy + \frac{1}{2} \int_{\Omega_{e_i}} n E_x \frac{\partial \zeta_2}{\partial y} dx dy, \\ \int_{\Omega_{e_i}} \zeta_2 \frac{\partial n}{\partial x} \frac{\partial \phi}{\partial y} dx dy &= \frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 \phi \mathcal{O}_x n_y dx dy - \frac{1}{2} \int_{\Omega_{e_i}} \phi \mathcal{O}_x \frac{\partial \zeta_2}{\partial y} dx dy \\ &\quad + \frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 n E_y n_x dx dy - \frac{1}{2} \int_{\Omega_{e_i}} E_y n \frac{\partial \zeta_2}{\partial x} dx dy. \end{aligned} \right. \quad (3.8)$$

In the vorticity equation, the first term on the RHS of (3.5) can be split up in the same way as the cross product terms from the density equation (3.6), where again the vector splits into its components $\mathcal{U} = (\mathcal{U}_x, \mathcal{U}_y)$. Again we can choose the quasistrong representation:

$$\underline{\text{The quasistrong flux}} : \left\{ \int_{\Omega_{e_i}} \zeta_2^i (\nabla_{\perp} \phi \times \nabla_{\perp} U) dx dy = \int_{\Omega_{e_i}} \zeta_2^i (\mathbf{E}_{\perp} \times \mathcal{U}) dx dy, \quad (3.9) \right.$$

that utilized the linearized variables for convection transport. Similarly, as with the mass equation, the strong case follows here as well

$$\underline{\text{The strong flux}} : \left\{ \begin{aligned} - \int_{\Omega_{e_i}} \zeta_2 \frac{\partial \phi}{\partial x} \frac{\partial U}{\partial y} dx dy &= -\frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 \phi \mathcal{U}_y n_x dx dy + \frac{1}{2} \int_{\Omega_{e_i}} \phi \mathcal{U}_y \frac{\partial \zeta_2}{\partial x} dx dy \\ &\quad - \frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 U \mathcal{O}_x n_y dx dy + \frac{1}{2} \int_{\Omega_{e_i}} U \mathcal{O}_x \frac{\partial \zeta_2}{\partial y} dx dy, \\ \int_{\Omega_{e_i}} \zeta_2 \frac{\partial U}{\partial x} \frac{\partial \phi}{\partial y} dx dy &= \frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 \phi \mathcal{U}_x n_y dx dy - \frac{1}{2} \int_{\Omega_{e_i}} \phi \mathcal{U}_x \frac{\partial \zeta_2}{\partial y} dx dy \\ &\quad + \frac{1}{2} \int_{\partial \Omega_{e_i}} \zeta_2 E_y U n_x dx dy - \frac{1}{2} \int_{\Omega_{e_i}} E_y U \frac{\partial \zeta_2}{\partial x} dx dy. \end{aligned} \right. \quad (3.10)$$

The major difference in the vorticity equation is the inclusion of the curvature term, $\beta \partial_y \ln n$. Here again we are presented with a choice of convective flux in the upwinding. Phenomenologically it is often that case that the scaling of the β curvature flux is small relative to the ion polarization flux. As a consequence the semistrong flux can be written:

$$\underline{\text{The semistrong flux}} : \left\{ - \int_{\Omega_{e_i}} \zeta_2^i \beta (\nabla_{\perp} x \times n^{-1} \nabla_{\perp} n) dx dy = - \int_{\Omega_{e_i}} \zeta_2^i \beta (\nabla_{\perp} x \times n^{-1} \mathcal{O}) dx dy. \quad (3.11) \right.$$

By semistrong we mean that this weaker version of the curvature term (3.11) is coupled to the stronger version of the ion polarization term (3.10). When the quasistrong flux is used, the entirety of the flux system is characterized as quasistrong. However, in this context we can again restrict to the completely strong flux form:

$$\underline{\text{The strong flux}} : \left\{ - \int_{\Omega_{e_i}} \beta \zeta_2 \frac{\partial \ln n}{\partial y} dx dy = - \int_{\partial \Omega_{e_i}} \beta \ln n \zeta_2 n_y dx dy + \int_{\Omega_{e_i}} \beta \ln n \frac{\partial \zeta_2}{\partial y} dx dy. \right. \quad (3.12)$$

We will show some numerical results for these different choice of fluxes below.

3.1.4 The elliptic subsystems

As it turns out, solving elliptic subsystems (such as the Poisson equation) is somewhat more involved than solving convection dominated subsystems, as elliptic problems for example require solving a global linear system numerically. Thus we will use the Poisson subsystem to demonstrate how the numerical fluxes can be chosen in order to form a well-conditioned linear system.

First note that (3.3) can be written as,

$$\begin{aligned} \int_{\partial \Omega_{e_i}} \zeta_3^i \mathbf{E}_\perp \cdot \mathbf{n} dx dy - \int_{\Omega_{e_i}} \mathbf{E}_\perp \cdot \nabla_\perp \zeta_3^i dx dy &= \int_{\Omega_{e_i}} \zeta_3^i U dx dy, \\ \int_{\Omega_{e_i}} \mathbf{E}_\perp \cdot \xi_3^i dx dy &= \int_{\partial \Omega_{e_i}} \xi_3^i \cdot \mathbf{n} \phi dx dy - \int_{\Omega_{e_i}} \phi \nabla_\perp \cdot \xi_3^i dx dy, \end{aligned} \quad (3.13)$$

where the second equation provides the solution to the electric field. Approximating the first equation of (3.13) componentwise, we find that the first term on the LHS and the source term on the RHS are each approximated in the usual way:

$$\int_{\partial \Omega_{e_i}} \mathbf{E}_\perp \cdot \nabla_\perp \zeta_3^i dx dy \approx \int_{\Omega_{e_i}} (\mathbf{E}_\perp)_{hp}^i \cdot \nabla_\perp \zeta_3^i dx dy \quad \text{and} \quad \int_{\Omega_{e_i}} \zeta_3 U dx dy \approx \int_{\Omega_{e_i}} \zeta_3 U_{hp}^i dx dy.$$

The remaining term leads to another numerical flux $\hat{\mathfrak{E}}_i$, such that

$$\begin{aligned} \mathfrak{E}_i((\mathbf{E}_\perp)_{hp}^i, \zeta_3^i) &= \sum_{j \in \Xi(i)} \int_{\Gamma_{ij}} \hat{\mathfrak{E}}_i((\mathbf{E}_\perp)_{hp}^i|_{\Gamma_{ij}}, (\mathbf{E}_\perp)_{hp}^i|_{\Gamma_{ji}}) \cdot \mathbf{n}_{ij} \zeta_3^i|_{\Gamma_{ij}} dx dy \\ &\approx \sum_{j \in \Xi(i)} \int_{\Gamma_{ij}} \sum_{l=1}^{\dim} ((\mathbf{E}_\perp)_{hp}^i)_l (n_{ij})_l \zeta_3^i|_{\Gamma_{ij}} d\Xi. \end{aligned}$$

Likewise the first and last terms in the auxiliary equation of (3.13) are approximated using,

$$\int_{\Omega_{e_i}} \mathbf{E}_\perp \cdot \xi_3^i dx dy \approx \int_{\Omega_{e_i}} (\mathbf{E}_\perp)_{hp}^i \cdot \xi_3^i dx dy \quad \text{and} \quad \int_{\Omega_{e_i}} \phi \nabla_\perp \cdot \xi_3^i dx dy \approx \int_{\Omega_{e_i}} \phi_{hp}^i \nabla_\perp \cdot \xi_3^i dx dy$$

where the numerical flux $\hat{\mathcal{T}}_i$ becomes:

$$\begin{aligned} \mathcal{T}_i(\theta_{hp}^i, \xi_3^i) &= \sum_{j \in \Xi(i)} \int_{\Gamma_{ij}} \hat{\mathcal{T}}_i(\theta_{hp}^i|_{\Gamma_{ij}}, \theta_{hp}^i|_{\Gamma_{ji}}) \mathbf{n}_{ij} \cdot \xi_3^i|_{\Gamma_{ij}} dx dy \\ &\approx \sum_{j \in \Xi(i)} \int_{\Gamma_{ij}} \sum_{l=1}^{\dim} (\theta_{hp}^i)_l|_{\Gamma_{ij}} (n_{ij})_l (\xi_3^i)_l d\Xi. \end{aligned} \quad (3.14)$$

First we define the *jump* $[[\cdot]]$ and *average* $\{\cdot\}$ for a vector \mathbf{v} on the shared interior faces as,

$$[[\mathbf{v}]] = \mathbf{v}_1 \cdot \mathbf{n}_1 + \mathbf{v}_2 \cdot \mathbf{n}_2, \quad \text{and} \quad \{\mathbf{v}\} = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2).$$

If we further let the union of the discrete boundaries of the domain be denoted by Γ and define the edge values without the physical boundary as $\Gamma^0 = \Gamma \setminus \partial\Omega$, then using this notation, it is a direct computation to check that we can rewrite our fluxes, for example (3.14), in the form:

$$\begin{aligned} \mathcal{T}_i(\theta_{hp}^i, \xi_3^i) &= \sum_{j \in \Xi(i)} \int_{\Gamma_{ij}} \hat{\mathcal{T}}_i(\theta_{hp}^i|_{\Gamma_{ij}}, \theta_{hp}^i|_{\Gamma_{ji}}) \mathbf{n}_{ij} \cdot \xi_3^i|_{\Gamma_{ij}} dx dy \\ &= \int_{\Gamma} [[\hat{\mathcal{T}}_i]] \cdot \{\xi_3^i\} dx dy + \int_{\Gamma^0} \{\hat{\mathcal{T}}_i\} [[\xi_3^i]] dx dy. \end{aligned} \quad (3.15)$$

If we use this form on both the numerical fluxes in (3.13) and perform another integration by parts on the last term in the auxiliary equation, we have that the full auxiliary equation can be written:

$$\int_{\Omega_{e_i}} \mathbf{E}_{\perp} \cdot \xi_3^i dx dy = \int_{\Gamma} [[\hat{\mathcal{T}}_i - \phi_{hp}^i]] \cdot \{\xi_3^i\} dx dy + \int_{\Gamma^0} \{\hat{\mathcal{T}}_i - \phi_{hp}^i\} [[\xi_3^i]] dx dy + \int_{\Omega_{e_i}} \xi_3^i \cdot \nabla_{\perp} \phi_{hp}^i dx dy. \quad (3.16)$$

Now we define the *lifting operators* over the shared edges Γ , as

$$\int_{\Omega_e} r(v) \cdot \xi dx dy = - \int_{\Gamma} v \cdot \{\xi\} dx dy, \quad \text{and} \quad \int_{\Omega_e} l(w) \cdot \xi dx dy = - \int_{\Gamma^0} w [[\xi]] dx dy$$

such that (3.16) can be rewritten in the primal formulation as,

$$\int_{\Omega_{e_i}} \mathbf{E}_{\perp} \cdot \xi_3^i dx dy = \int_{\Omega_{e_i}} \xi_3^i \cdot \nabla_{\perp} \phi_{hp}^i dx dy - \int_{\Gamma} r([[\hat{\mathcal{T}}_i - \phi_{hp}^i]]) \xi_3^i dx dy - \int_{\Gamma^0} l(\{\hat{\mathcal{T}}_i - \phi_{hp}^i\}) \xi_3^i dx dy. \quad (3.17)$$

Applying these same operations to the equations in (3.13), where we substitute (3.16) into the first equation of (3.13) by substituting for ξ_3^i the term $\nabla_{\perp} \zeta_3^i$, then we arrive with the following bilinear primal form of the first equation,

$$\begin{aligned} \int_{\Omega_{e_i}} \mathbf{E}_{\perp} \cdot \nabla_{\perp} \zeta_3^i dx dy + \int_{\Gamma} \left([[\hat{\mathcal{T}}_i - \phi_{hp}^i]] \cdot \{\nabla_{\perp} \zeta_3^i\} - \{\hat{\mathcal{E}}_i\} [[\zeta_3^i]] \right) dx dy \\ + \int_{\Gamma^0} \left(\{\hat{\mathcal{T}}_i - \phi_{hp}^i\} [[\nabla_{\perp} \zeta_3^i]] - [[\hat{\mathcal{E}}_i]] \{\zeta_3^i\} \right) dx dy = - \int_{\Omega_{e_i}} \zeta_3^i U dx dy \end{aligned} \quad (3.18)$$

It is this formulation (3.18) and (3.17) from which arises the ‘‘choice’’ of numerical fluxes in parabolic and elliptic subsystems. There are large classes of such fluxes to choose from, many of which have been studied in depth with specific aims in mind, such as particular advantages in stability, efficiency, and/or rates of convergence. One of the more robust of these methods is the classic interior penalty method [3]. The classical choice of fluxes for the IPG method corresponding to the Poisson equation would be

$$\hat{\mathcal{T}}_i = \{\phi_{hp}^i\} \text{ on } \Gamma^0, \quad \hat{\mathcal{T}}_i = 0 \text{ on } \partial\Omega, \quad \text{and} \quad \hat{\mathcal{E}}_i = \{\nabla_{\perp} \phi_{hp}^i\} - \sigma [[\phi_{hp}^i]] \text{ on } \Gamma.$$

Plugging these fluxes in (3.18) and (3.17) then directly yields,

$$\int_{\Omega_{e_i}} \mathbf{E}_\perp \cdot \nabla_\perp \zeta_3^i dx dy - \int_\Gamma (\llbracket \phi_{hp}^i \rrbracket \cdot \{\nabla_\perp \zeta_3^i\} + \{\mathbf{E}_\perp\} \cdot \llbracket \zeta_3^i \rrbracket + \sigma_i \llbracket \phi_{hp}^i \rrbracket \llbracket \zeta_3^i \rrbracket) dx dy = - \int_{\Omega_{e_i}} \zeta_3^i U dx dy, \quad (3.19)$$

where σ_i is the penalty parameter.

More generally, we can rewrite (3.19) in terms of a parameter ϱ ,

$$\int_{\Omega_{e_i}} \mathbf{E}_\perp \cdot \nabla_\perp \zeta_3^i dx dy - \int_\Gamma (\llbracket \phi_{hp}^i \rrbracket \cdot \{\nabla_\perp \zeta_3^i\} + \varrho \{\mathbf{E}_\perp\} \cdot \llbracket \zeta_3^i \rrbracket + \sigma_i \llbracket \phi_{hp}^i \rrbracket \llbracket \zeta_3^i \rrbracket) dx dy = - \int_{\Omega_{e_i}} \zeta_3^i U dx dy, \quad (3.20)$$

such that $\varrho = 1$ gives the classic IPG method (or symmetric interior penalty method, SIPG), $\varrho = 0$ gives the incomplete interior penalty method (IIPG) [18], and $\varrho = -1$ gives the nonsymmetric interior penalty method (NIPG) [58]. Also note that whenever appropriate σ_i is chosen to satisfy the superpenalization property [60] in order to improve the convergence order.

Notice thus far that this only provides a solution up to the boundary $\partial\Omega$ terms. To recover the nonperiodic boundary forcing we apply Nitsche's weak boundary conditions, setting

$$- \int_{\partial\Omega_{e_b}} \zeta_3^b \frac{\partial \phi_{hp}^b}{\partial \mathbf{n}} dx dy = \int_{\partial\Omega_{e_b}} \sigma_b \zeta_3^b (\phi_{hp,b}^b - \phi_{hp}^b) dx dy, \quad (3.21)$$

and integrating by parts. In this way we recover the total linear system for the initial-boundary problem (3.15), (3.17), and (3.21), which we rewrite as:

$$\begin{aligned} & \int_{\Omega_{e_i}} \mathbf{E}_\perp \cdot \nabla_\perp \zeta_3^i dx dy - \int_\Gamma (\llbracket \phi_{hp}^i \rrbracket \cdot \{\nabla_\perp \zeta_3^i\} + \varrho \{\mathbf{E}_\perp\} \cdot \llbracket \zeta_3^i \rrbracket + \sigma_i \llbracket \phi_{hp}^i \rrbracket \llbracket \zeta_3^i \rrbracket) dx dy \\ & - \int_{\partial\Omega_{e_b}} \left\{ \frac{\partial}{\partial \mathbf{n}} (\zeta_3^b \phi_{hp}^b) - \phi_{hp}^b \frac{\partial \zeta_3^b}{\partial \mathbf{n}} \right\} dx dy = - \int_{\Omega_{e_i}} \zeta_3^i U dx dy + \int_{\partial\Omega_{e_b}} \sigma_b \zeta_3^b (\phi_{hp,b}^b - \phi_{hp}^b) dx dy. \end{aligned} \quad (3.22)$$

The above formulation is referred to in the literature as the primal formulation for an elliptic system, and is essential for constructing a stiffness matrix. However, the fluxes arising in the parabolic subsystems of the convection dominated flows above (i.e. the diffusion component of the mass and vorticity conservation equations) can be directly obtained in a time-explicit solver simply by inserting the fluxes. For this purpose, we employ the consistent and stable Brezzi flux [11]

$$\hat{\mathcal{T}}_i = \{\phi_{hp}^i\} \text{ on } \Gamma^0, \quad \hat{\mathcal{T}}_i = 0 \text{ on } \partial\Omega, \quad \text{and} \quad \hat{\mathcal{E}}_i = \{(\mathbf{E}_\perp)_{hp}^i\} - \sigma \llbracket \phi_{hp}^i \rrbracket \text{ on } \Gamma,$$

where the penalty is a positive constant $\sigma > 0$.

3.2 The formulation

The semi-discrete form of the solution can be achieved by collecting the terms above. First define the state vectors $V_{hp}^i = (n_{hp}^i, \phi_{hp}^i)^\top$, $\zeta^i = (\zeta_1^i, \zeta_2^i)^\top$ and $\Sigma_{hp}^i = (\mathcal{O}_{hp}^i, \mathcal{U}_{hp}^i)^\top$, $\xi^i = (\xi_1^i, \xi_2^i)^\top$. Now define the product terms by setting

$$\begin{aligned} \chi_{hp}^i &= ((\mathbf{E}_\perp)_{hp} \times \mathcal{O}_{hp}^i, (\mathbf{E}_\perp)_{hp}^i \times \mathcal{U}_{hp}^i) + \beta ((\nabla_\perp x \times (n_{hp}^i)^{-1} \mathcal{O}_{hp}^i))^\top, \\ \gamma_{hp}^i &= (\alpha n_{hp}^i, -\alpha \phi_{hp}^i)^\top, \quad \text{and} \quad \mathcal{S}_{hp}^i = (D \mathcal{O}_{hp}^i, \mu \mathcal{U}_{hp}^i)^\top, \end{aligned}$$

the numerical fluxes can be set such that $\mathcal{F}_{hp}^i = (\mathcal{D}_i, \mathcal{M}_i)^\top$ and $\mathcal{A}_{hp} = (\mathcal{N}_i, \mathcal{Q}_i)^\top$, and we further define:

$$\Gamma_{\mathcal{G}} = \int_{\Gamma} (\llbracket \phi_{hp}^i \rrbracket \cdot \{\nabla_{\perp} \zeta_3^i\} + \varrho \{\mathbf{E}_{\perp}\} \cdot \llbracket \zeta_3^i \rrbracket + \sigma_i \llbracket \phi_{hp}^i \rrbracket \llbracket \zeta_3^i \rrbracket) dx dy,$$

Then setting $\mathcal{X}_{hp} = \sum_{\Omega_{e_i} \in \mathcal{T}_h} \mathcal{X}_{hp}^i$, and letting the generalized scalar product

$$(a_{hp}, b_{hp})_{\Omega_{\mathcal{G}}} = \sum_{\Omega_{e_i} \in \mathcal{T}_{hp}} \int_{\Omega_{e_i}} a_{hp}^i \cdot b_{hp} dx dy,$$

we can write our semidiscrete approximate solution to (2.1)-(2.2) as the pair of functions (V_{hp}, Σ_{hp}) for all $t \in (0, T)$ satisfying:

The semidiscrete discontinuous Galerkin formulation

- a) $V_{hp} \in C^1([0, T]; S_h^p)$, $\Sigma_{hp} \in S_h^p$,
 - b) $V_{hp}(0) = \Pi_{hp} V_0$,
 - c) $(\mathcal{U}_{hp}, \xi)_{\Omega_{\mathcal{G}}} = \mathcal{A}_{hp} - (U_{hp}, \nabla_{\perp} \cdot \xi)_{\Omega_{\mathcal{G}}}$,
 - d) $\frac{d}{dt} (V_{hp}, \zeta)_{\Omega_{\mathcal{G}}} = \mathcal{F}_{hp} - (\chi_{hp}, \zeta)_{\Omega_{\mathcal{G}}} - (\gamma_{hp}, \zeta)_{\Omega_{\mathcal{G}}} - (\mathcal{J}_{hp}, \nabla_{\perp} \zeta)_{\Omega_{\mathcal{G}}}$,
 - e) $(\mathbf{E}_{\perp}, \nabla_{\perp} \zeta_3)_{\Omega_{\mathcal{G}}} - \Gamma_{\mathcal{G}} = -(\zeta_3, U_{hp})_{\Omega_{\mathcal{G}}} + \text{boundary terms}$
- (3.23)

where Π_{hp} is the projection/interpolation operator onto the space of discontinuous piecewise polynomials S_h^p . In the modal basis we utilize the standard L^2 -projection, given for a function $f_0 \in L^2(\Omega_{e_i})$ such that our approximate projection $f_{0,h} \in L^2(\Omega_{e_i})$ is obtained by solving, $\int_{\Omega_{e_i}} f_{0,h} \zeta dx = \int_{\Omega_{e_i}} f_0 \zeta dx$. In the nodal DG setting, on the other hand, we use the interpolation operator.

The discretization in time follows now directly from (3.23), where we employ the family of SSP (strong stability preserving, or often ‘‘total variation diminishing (TVD)’’) Runge–Kutta schemes discussed in [36, 38, 63, 66]. These schemes are chosen as (3.23) is a convection dominated evolution equation, and thus the above SSPRK schemes are well-suited for enhancing the stability regions of the associated spectrum of the hyperbolic operators (in the usual sense of linear von Neumann analysis). In the case of reaction-type or diffusion-type dominated system, the RKSSP family chosen is not necessarily ideal, which is directly analyzed in a concurrent work [51].

§4 Numerical tests

The numerical residual e_h for any solution component (e.g. we use ϕ here) is given by $e_h = \phi - \phi_{hp}$, where ϕ is the solution to (2.3) and ϕ_{hp} its discrete approximate counterpart. This measure will be enough to determine the convergence order and many aspects of the numerical behavior of the solution relative to its analytic counterpart. Notice that since ArcOn utilizes both modal and nodal basis functions, there is a potential ambiguity that arises in the error measures. Namely, the initial state of the system $\phi_{hp}|_{t=0}$ must be either projected or interpolated into the appropriate discrete basis. This projection error in the modal DG case, or interpolation error in the nodal DG case, is frequently nonzero (e.g. always nonzero for analytic functions), and has a convergence order of its own relative to the exact solution.

For the remainder of the paper we will restrict to the nodal DG case, and focus rather on the model error and not the way in which the truncation error distributes about the representation space. In

this sense, since we expect our variational strong solutions in the L^2 -norm to inherit an inner product space, we will concern ourselves with the L^2 -norm of the model error,

$$\|e_h\|_{L^2(\Omega_h)} = \left(\int_{\Omega_h} e_h^2 dx dy \right)^{1/2}.$$

Without having performed the error analysis on (2.1)–(2.3), we assume (as is conventional) that given smooth initial data with well-constrained boundary behavior, the solution converges at the optimal error rate $\|e\|_{L^2(\Omega_h)} \leq Ch^{p+1}$. We can estimate the error rate in our numerical tests \mathcal{C} by computing the slope,

$$\mathcal{C} = \frac{1}{\ln(2)} \ln \left(\frac{\|e_h\|_{L^2(\Omega_h)}}{\|e_{h/2}\|_{L^2(\Omega_h)}} \right),$$

where this measure \mathcal{C} is characterized as the convergence rate of the method.

4.1 Elliptic verification

Let us proceed by demonstrating the numerical convergence of the components of our system for some numerically well-behaved problems. First we begin with the elliptic solver. Consider the Poisson equation with Dirichlet boundary data

$$U = \nabla_{\perp}^2 \phi, \quad \phi_b = U_b = 0 \text{ on } \partial\Omega.$$

A solution is chosen so as to satisfy

$$\phi = 1 \times 10^{-2} e^{(-x^2/2 - y^2/2)}$$

over the domain $x \in [-30, 30], y \in [-20, 20]$.

The mesh is chosen to be uniform and quadrangular, where for the sake of brevity, we utilize the classic IP method (SIPG) here. Here the stiffness matrix is preconditioned using a BoomerAMG algebraic multigrid preconditioner from the LLNL HYPRE suite coupled through PETSc. The convergence threshold is set to 1×10^{-12} and superpenalization is used. The case was tested in parallel using between 16 and 96 processors, for twelve test cases with $p \in \{1, 2, 3\}$ and $h \in \{1/8, 1/16, 1/32, 1/64\}$. Note that h here is set relative to d_y which has a $1 : 3/2$ aspect ratio to d_x ; thus, $h = 64$ implies here that $d_y/64$ and $d_x/96$, etc. The finite element space used here were nodal Lagrange interpolating polynomials with support points at the Gauss-Legendre quadrature points. The resulting convergence rates and errors are provided in Table 1, indicating the expected behavior.

The graphs in figure 1 are provided to help develop an intuition for the error weighting across a Gaussian type solution in the local spatially spectral basis. It should be duly noted here that in the SIPG method, e_h is highly dependent on the penalty term σ . In the graphs of figure 1 the penalty term is held constant $\sigma = 1000$, and the error residual e_h is plotted; hence it is important to realize that these do not represent fully convergent cases.

4.2 Hyperbolic-parabolic verification

Next we address the convergence properties of the explicit convection-diffusion problems. Here we consider the continuity equation:

$$\partial_t n + [\phi, n] + \alpha n - D \nabla_{\perp}^2 n = f, \quad n|_{t=0} = n_0, \quad n = n_b \text{ on } \partial\Omega, \quad (4.1)$$

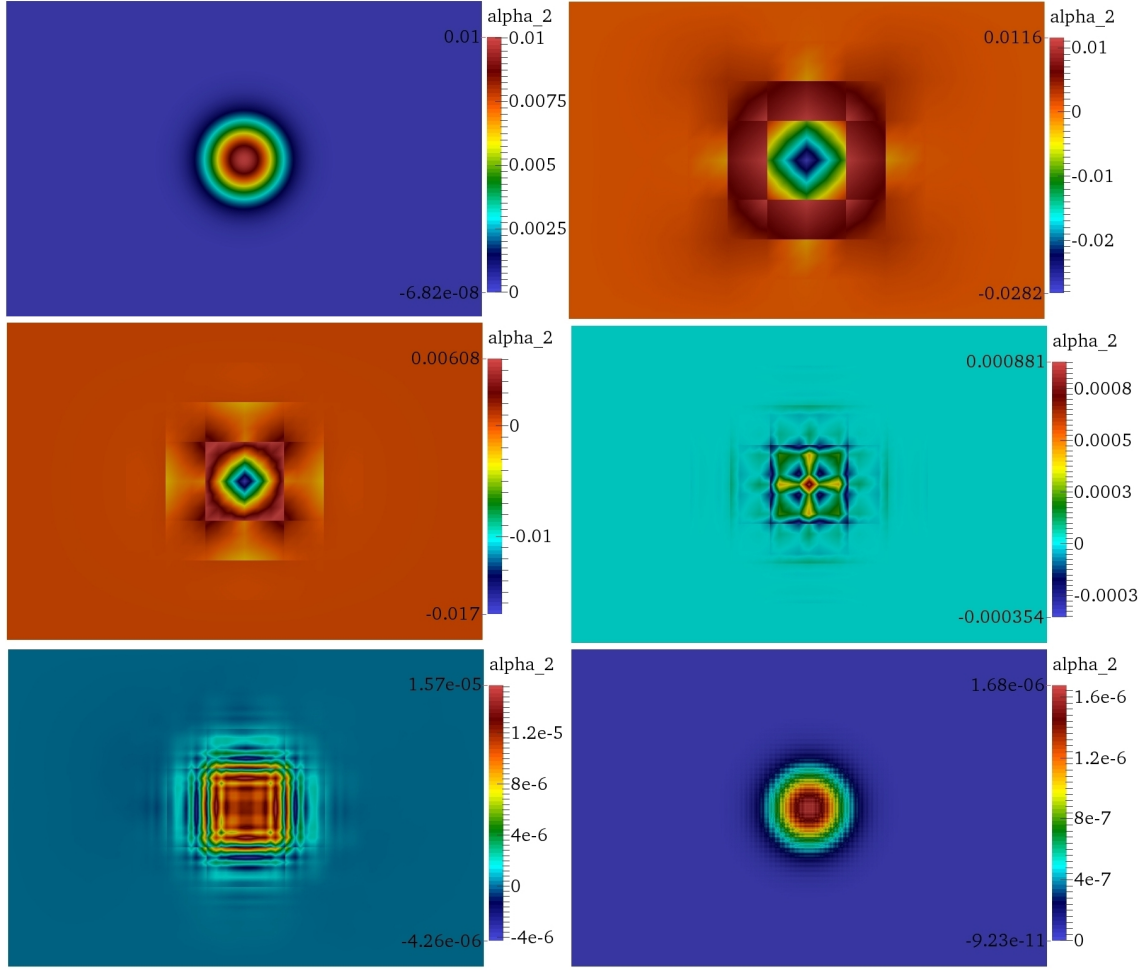


Figure 1: The residual $\alpha_2 = e_h$ of the Poisson convergence tests, in clockwise order from top, exact solution, $(p = 1, h = d_x/12, d_y/8)$, $(p = 2, h = d_x/12, d_y/8)$, $(p = 3, h = d_x/12, d_y/8)$, $(p = 7, h = d_x/12, d_y/8)$, and $(p = 3, h = d_x/96, d_y/64)$.

p	σ	$L^2(\Omega_h)$	h	\mathcal{C}
1	$\frac{13}{4}$	7.46285×10^{-3}	$d_x/96, d_y/64$	2.0483
2	12	7.583588×10^{-4}	$d_x/96, d_y/64$	3.0512
3	90	8.228863×10^{-5}	$d_x/96, d_y/64$	4.0376

Table 1: The L^2 -error and convergence behavior of the nodal DG solution for the Poisson solver. The distance $d_x = 60$ and $d_y = 40$.

and use the standard manufactured solutions method. We set,

$$n = te^{-(x^2+y^2)/2}, \quad \phi = \gamma \cos(x) \cos(y).$$

p	γ, D	$L^2(\Omega_h)$	h	\mathcal{E}
1	0.1, 0.01	3.07778×10^{-2}	$\pi/16$	2.5772
2	0.1, 0.01	1.33313×10^{-2}	$\pi/16$	3.0362
3	0.1, 0.01	7.48107×10^{-5}	$\pi/16$	4.3602

Table 2: The L^2 -error and convergence behavior of the nodal DG solution for the continuity equation. To completely quench CFL related errors and temporal error, the timestep taken is $dt = 1 \times 10^{-4}$ with SSPRK(6,4). Errors are taken to 200 timesteps, or $T = 0.02$.

The reaction term, when viewed through the lens of a fractional multistep method, is simply a linear first order ordinary differential equation whose numerical stiffness can be scaled by α , as $|\alpha|$ is the *stiffness index*. As such, it is fairly well-understand, and thus not the focus of our present study. Hence, we set $\alpha = 0$ for the convergence test. Let us note here that fast reaction modes may be split using operator splitting techniques, though this topic lies beyond the scope of this present review (see [52] for previous work in this area), and we can assume that diffusion and reactions modes operate on similar timescales for our present purposes.

The test domain is chosen as $\Omega = [-2\pi, 2\pi]^2$ with fully periodic boundary conditions, such that in order to satisfy the exact equation the manufactured source term must be set to:

$$f = (1 + tx\gamma \cos(x) \sin(y) - ty\gamma \sin(x) \cos(y) + 2Dt - Dtx^2 - Dty^2) e^{-(x^2-y^2)/2},$$

where $\gamma = 0.1$ and $D = 0.01$. These results are notated in Table 2 where we are pleased to observe that in this particular instance, ArcOn is demonstrating superconvergence in h for p odd. This is notable as we have set the penalty σ from the Brezzi flux to machine epsilon, effectively reducing to a standard Bassi-Rebay type flux [8], which has good convergence properties but may want for stability when employed for highly nonlinear applications. Also we have set the convective fluxes here to the strong flux form (3.8).

For the vorticity equation we perform a similar periodic test case,

$$\partial_t U + [\phi, U] - \alpha\phi + \beta[x, \ln n] - \mu \nabla_{\perp}^2 U = g \quad U|_{t=0} = U_0, \quad U = U_b \text{ on } \partial\Omega,$$

where using the same rational as above $\alpha = 0$. In this case the following exact forms are chosen for the three fields:

$$U = (t + 1)e^{(x^2+y^2)/2}, \quad \phi = \cos(x) \cos(y), \quad \ln n = \gamma e^{\operatorname{sech}(x)\operatorname{sech}y}.$$

Now the source term over the same physical domain Ω becomes:

$$g = f - \gamma \operatorname{sech}(x)\operatorname{sech}(y) \tanh(y) e^{\operatorname{sech}(x)\operatorname{sech}(y)}.$$

The convergence results are shown in Table 3 and Table 4 and run using equidistant nodal Lagrange polynomial spaces evaluated using Gauss-Lobatto cubature rules. What is most notable here is that the strong and the quasistrong fluxes demonstrate essentially equivalent convergence behavior for this test case. Even in figure 2 the error residual is indistinguishable between the two solutions. Further testing of these flux formulations has demonstrated that the quasistrong and semistrong fluxes have very similar behavior to the strong form in the standard settings We will return to this in slightly more detail in section 5.1.

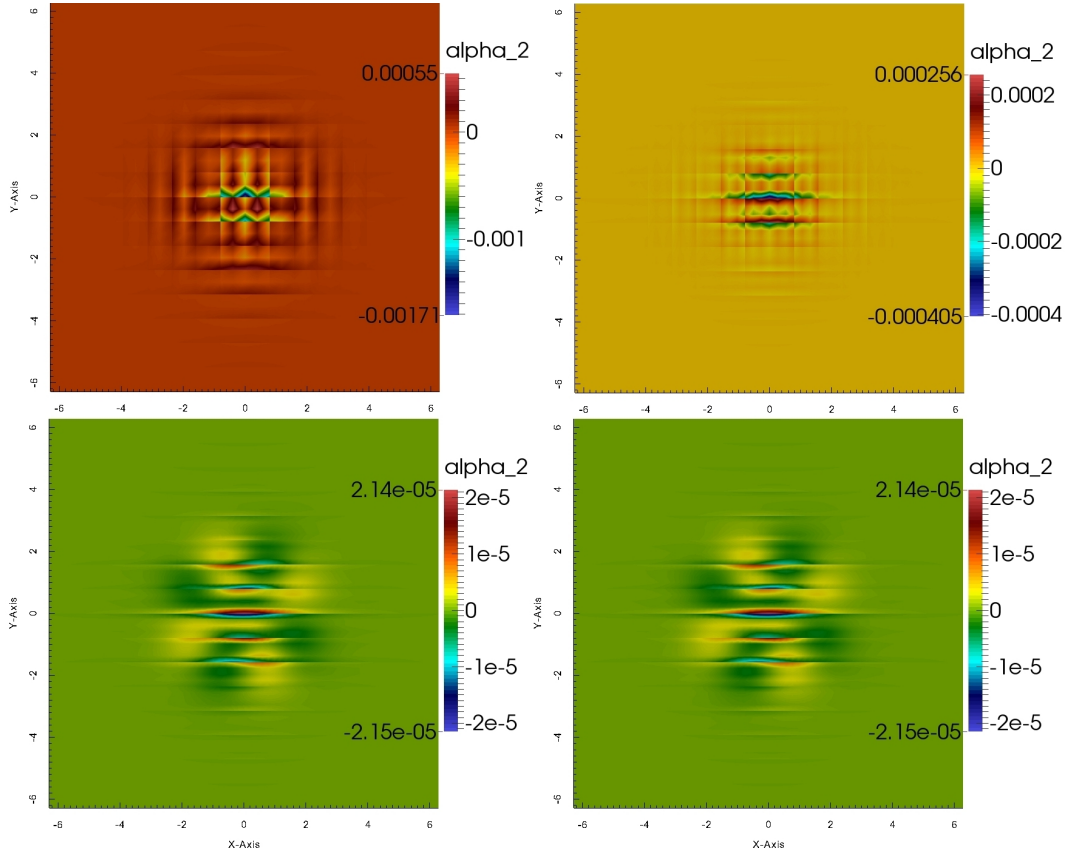


Figure 2: The residual $\alpha_2 = e_h$ of the vorticity convergence tests. In clockwise order from top left, $p = 1, h = \pi/16$ strong flux, $p = 2, h = \pi/16$ strong flux, $p = 3, h = \pi/16$ strong flux, and $p = 3, h = \pi/16$ quasistrong flux.

p	γ, D	$L^2(\Omega_h)$	h	\mathcal{C}
1	0.1, 0.01	6.21249×10^{-3}	$\pi/64$	2.3311
2	0.1, 0.01	1.57556×10^{-3}	$\pi/64$	3.5003
3	0.1, 0.01	7.48107×10^{-5}	$\pi/64$	4.9782

Table 3: The L^2 -error and convergence behavior of the nodal DG solution for the vorticity equation using the strong fluxes (3.10) and (3.12). As above the timestep taken is $dt = 1 \times 10^{-4}$ with SSPRK(6,4) and $T = 0.015$.

4.3 Streamer validation

A simple physically relevant case to test the system on, is blob formation in the scrape-off layer of tokamaks. Here we set the domain to $\Omega_h = [-7, 7] \times [-12, 12]$, setting periodic boundary conditions. The initial state of the system scaled to dimensionless units [5] is:

$$n_0 = e^{-(x^2+y^2)/2} \quad \text{and} \quad \phi_0 = U_0 = 0.$$

p	γ, D	$L^2(\Omega_h)$	h	\mathcal{E}
1	0.1, 0.01	6.24371×10^{-3}	$\pi/64$	2.3311
2	0.1, 0.01	1.56755×10^{-3}	$\pi/64$	3.5077
3	0.1, 0.01	1.28545×10^{-5}	$\pi/64$	4.9705

Table 4: The L^2 -error and convergence behavior of the nodal DG solution for the vorticity equation using the quasistrong fluxes (3.9) and (3.11). As above the timestep taken is $dt = 1 \times 10^{-4}$ with SSPRK(6,4) and $T = 0.015$.

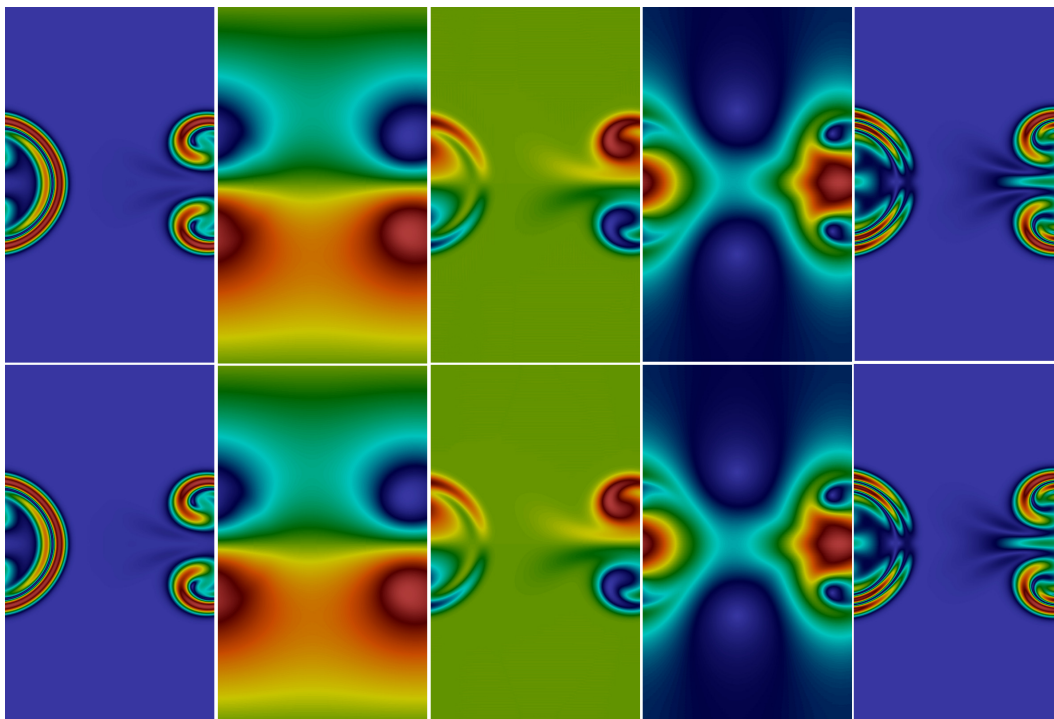


Figure 3: Here we see the SOL streamer with $D = \mu = 3 \times 10^{-2}$ at $T = 13.2$. The top images show the strong flux solution, and the bottom show the quasistrong solution. In order the variables shown from left to right are: $\theta, \phi, U, \mathbf{E}_\perp$, and \mathcal{U} .

The boundaries conditions are periodic in x , and transmissive (or radiative) in y , meaning that the solution on the outer face is set equal to the value taken on the inner face at the boundary.

We have benchmarked this traveling blob solution as shown in figure 10 against BOUT++ [23] and Garcia, Bian, and Fundamenski [26], and found (as similarly discussed by Aydemir [5] though in a different context, see Fig. 5) that the formation of sharp gradients in the leading blob edge portend the formation of instabilities. In the low viscosity/diffusion limits (e.g. $Ra < 10^{-8}$) artificial diffusion as discussed in section 5, can be used to dampen these effects, and done so with very satisfying results.

In figure 3 we show a slightly different view of the solution, comparing various numerical features. In this test, as in the remainder of the paper, we use a discontinuous tensor product of Lagrange

polynomials with arbitrary nodal support points in order to preserve the diagonal mass matrix. Here $\alpha = \alpha_0 = 0$ and $\beta = 1$. We use 8192 quadrangular elements, $p = 3$ giving 1179648 degrees of freedom. The elliptic subsystem was solved using SIPG with the same settings from section 4.1, given the penalty parameter $\sigma = 75$, and the variable timestepper described in section 5 is also used. All tests were run on 64 processors.

In figure 3 some of the aspects of the solution are shown, where again it is clear that little difference exists between the strong and the quasistrong fluxes. The nuance that exists between the different flux forms can be found in the semistrong flux relative to the formulation of the system of equations. Namely, when (2.1)-(2.3) is solved relative to the n -independent toroidal curvature term β . In this case, given an absence of scale separation in the background plasma density, the leading term in (3.11) becomes unmanageable in the semistrong flux, and the only alternative is to use a numerical cutoff around the vacuum in the strong flux form. However, when the curvature term is reformulated, then the semistrong flux shows similar behavior to the strong and quasistrong fluxes, since then it is merely a mixture between the two. It is also worth noting that preliminary numerical tests suggest that the strong form may in some cases be better suited for numerical runs utilizing the regularization techniques discussed in section 5, in order to recover solutions in stable parameter spaces under more amenable timestepping constraints. The quasistrong and semistrong fluxes in the linear case, however, show better stability and performance features as the diffusion coefficients μ and D are increased. This observation indicates a subtle and delicate interplay between the flux forms when coupled with regularization techniques, and requires careful and further analysis, particularly as the regularization techniques require tuning parameters. Finally a notable feature of the solution is the structural consistency in the vorticity U and gradient of the vorticity \mathcal{U} in figure 3. Indeed the sharp leading edge of the vorticity gradient can be a source of instability in blob simulations, and the recovery of these profiles indicates a noteworthy robustness in the numerical formulation.

§5 Regularization processes

As the system of equations (2.1)–(2.3) is convection-dominated, dispersive, gradient driven and non-local, it presents its own set of numerical complications and nuances. For example, the cross product terms induce convective nonlinearities that are driven by dispersive modes. As a result, in a strict mathematical sense the natural function space that support solutions to such a system is highly irregular, and in all likelihood local in its signature behavior (e.g. $L^p_{loc}(\Omega)$). While this bodes well for justifying the use of a method such as DG, which is able to probe such a space with high order accuracy, it also introduces instabilities into the space. There are a number of methods for regularizing a DG solution, some of the most popular of which have traditionally included slope or flux limiting methods [39–41], energy stabilization techniques [52, 54], and spectral filtering [55]. Here we regularize (2.1)–(2.3) by way of artificial diffusion, which is capable of preserving physical nonlinearities in the solution to higher resolution in p .

5.1 Artificial diffusion and gradient driven flow

A frequent source of concern in nonlinear numerics is the development of instabilities that pollute solution accuracy. In a DG method these instabilities are augmented by the fact that interelement jumps are usually nonzero, and thus whenever large gradients become dominant, numerical shocks can lead to local pollution of solution accuracy order [55]. A particularly nice solution to such behavior

is provided by the addition of so-called artificial diffusion/viscosity into the right hand side operator [71, 76]. The basic idea is to preserve solution nonlinearities by exploiting the high order behavior of the Laplace operator, while simultaneously stymieing local instability formation. This approach can be physically motivated in that the methodology is very closely related to the artificial viscosity techniques that arise in classical large eddy simulation models for fluids [10, 47], with the only significance difference being in the details of the smoothing algorithm, which is here tuned for (2.1)-(2.3). Like those models, the extra viscosity/diffusion that is added can be interpreted a priori as arising from fine scale dissipation effects due to locally large energy gradients, though this interpretation requires care and further study to be made precise, and to confirm that the affects of the smoothing operation can, or in some parameter regimes might be physically motivated.

Formulating the artificial diffusion/viscosity is achieved by supplementing the diffusion components of (2.1) and (2.2) with the slightly perturbed system

$$\partial_t n + [\phi, n] = -\alpha_0 n + (D + \varepsilon_1) \nabla_{\perp}^2 n, \quad n|_{t=0} = n_0, \quad n = n_b \text{ on } \partial\Omega, \quad (5.1)$$

$$\partial_t U + [\phi, U] = \alpha\phi - \beta[x, \ln n] + (\mu + \varepsilon_2) \nabla_{\perp}^2 U, \quad U|_{t=0} = U_0, \quad U = U_b \text{ on } \partial\Omega, \quad (5.2)$$

$$U = \nabla_{\perp}^2 \phi, \quad \phi|_{t=0} = \phi_0, \quad \phi = \phi_b \text{ on } \partial\Omega, \quad (5.3)$$

where the $\varepsilon_i = \varepsilon_i(t, \mathbf{x})$ are functions of space and time, and represent the stabilizing artificial diffusion/viscosity parameters.

These parameters are determined by using local regularity (or entropy) measures that are meant to sense the onset and formation of locally destabilizing numerical shock fronts. These sensors are often formed by way of modal truncation or through the use of mathematical entropies. In our case, since we are concerned with a gradient driven dispersive flow, we develop a regularity sensor that identifies large local variations in the solution by way of sensing the norm of the gradient over the global maximum. That is, for a vector \mathbf{v} we set the entropy sensor

$$S_{ij}(t, \mathbf{x}) = \left(\frac{L_{loc}^{\infty}(\nabla_{\perp} \mathbf{v}_h^i)}{\sup_{\Omega_h} \mathbf{v}_h} \right) \quad \text{for } \Omega_i \in \Omega_H \text{ and } j = 1, 2$$

where j runs over the indices of the state vector. Note that the gradient here is approximated in the mixed form here by the support variables $\mathcal{O}, \mathcal{U}, \mathbf{E}_{\perp}$.

This sensor is used to determine which cells Ω_i should be “regularized.” Each component of the artificial diffusion ε_j is calculated cell-wise Ω_i by setting

$$\varepsilon_{ij} = \begin{cases} 0, & \text{if } S_i < S_0 - \kappa \\ \frac{\varepsilon_i}{2} * \sin\left(\pi \left(\frac{S_i - S_0 + \kappa}{2\kappa}\right)\right), & \text{if } S_0 - \kappa \leq S_i < S_0 + \kappa \\ \varepsilon_i, & \text{if } S_i \geq S_0 + \kappa \end{cases}$$

Reasonably good starting points for the tuning parameters are $S_0 \sim \max_{\Omega_h} S(0, \mathbf{x})$, $\varepsilon \sim h/p$, and $\kappa \sim \text{Avg}_{\Omega_h} S(0, \mathbf{x})$.

In addition to the artificial regularization above, we implement a CFL-based variable timestepping algorithm. The basic algorithm is inspired by [19], where the timestep is approximated by a CFL-type constraint, though in [19] the variable timestepper is determined locally in space, meaning that subdomains have different timesteps. Here our variable timestepper varies only in time, not in space, and is based on a single spatially global estimate of the most restrictive CFL condition. That is, we

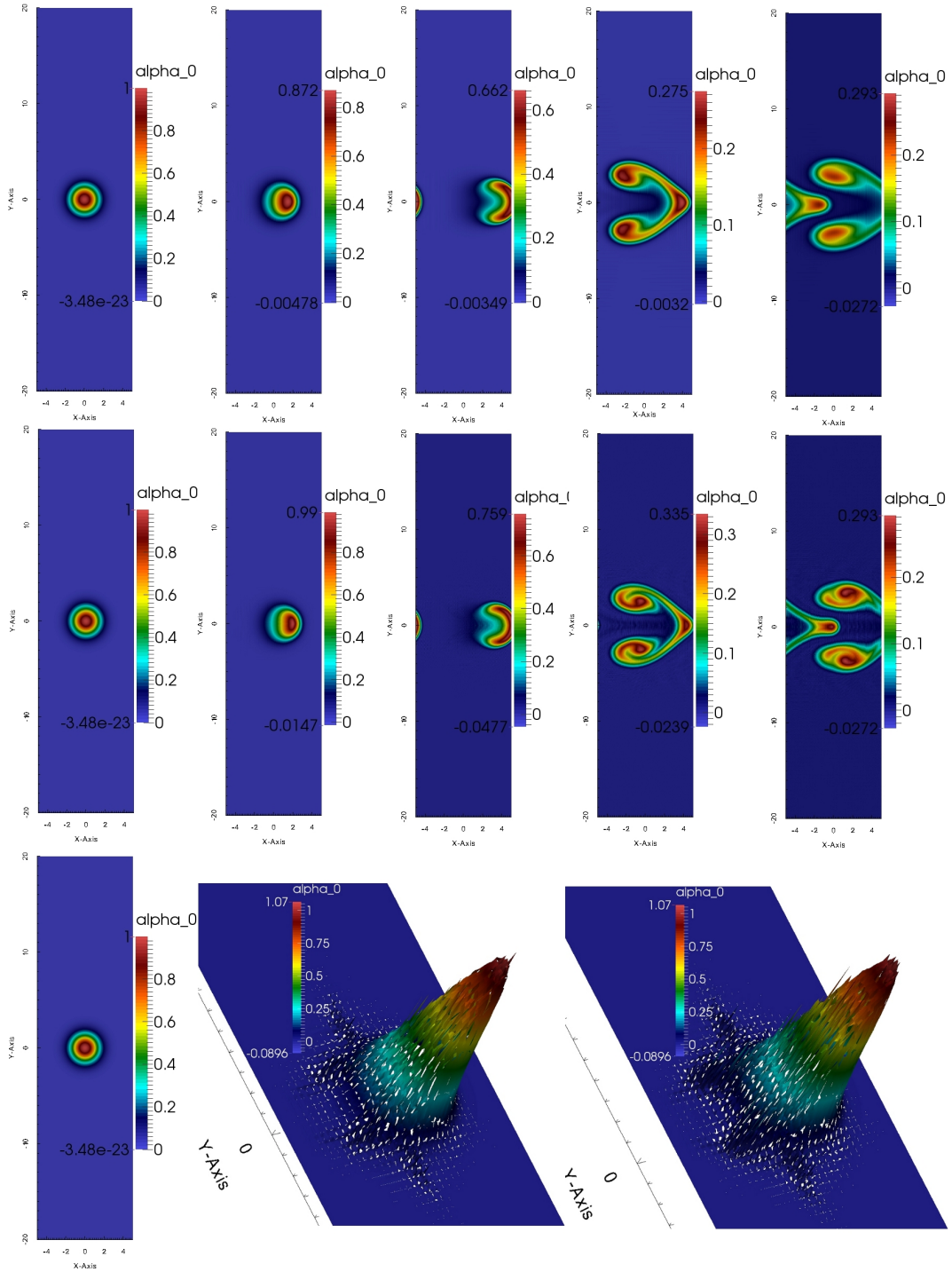


Figure 4: The top row is the SOL blob with $\varepsilon_i = 0, D = \mu = 3 \times 10^{-2}$, at the middle row with $\varepsilon_i = 3 \times 10^{-2}, D = \mu = 1 \times 10^{-8}, S_0 = 2, \kappa = 1.9$, and the bottom row is $\varepsilon_i = 0, D = \mu = 1 \times 10^{-8}$. See the text for which timesteps are being shown.

set $dt = \min_{\Omega_h} h/(2p+1)\mathfrak{C}$, where $\mathfrak{C}(t)$ is the maximum wavespeed at any timestep. This timestep is not optimal for $p > 2$ but does guarantee that the CFL constraint is maintained throughout [17, 69].

In figure 4 we show results from the streamer validation tests in section 4.3 using variable timestepping and setting $\Omega_h = [-5, 5] \times [-20, 20]$ where the y -direction is elongated to reduce boundary effects from the electric field. We solve this system over a small domain of 3072 uniform finite elements distributed across 64 processors. There are 3 components, with degree $p = 3$ polynomials, making 442368 total degrees of freedom in the computation. The wall clock time is approximately 5 minutes, running to 3600 timesteps. The top row of the figure shows the results with no artificial diffusion/viscosity, but with enough native diffusion/viscosity to maintain a stable solution; namely $D = \mu = 3 \times 10^{-2}$.

The five snapshots are at approximate times due to the variable timestepper, $T \approx 0, 3, 6, 14, 17$ where this corresponds to timesteps 0, 100, 300, 1000, 1300. Similar snapshots are shown in the second row where artificial diffusion/viscosity has been added, setting $\varepsilon_i = 3 \times 10^{-2}$, $D = \mu = 1 \times 10^{-8}$, $S_0 = 2$, $\kappa = 1.9$ corresponding to timesteps 0, 200, 600, 2300, 3000. Clearly the artificial regularization adds some expense to the computation in terms of computational complexity. The faster traveling modes in the lower diffusion case also restricts the CFL-based variable timestepper, leading to more total timesteps as well. It is important to note that adjusting κ and S_0 can have a fairly large impact on the solution behavior. Moreso, we see what we might qualitatively expect from a solution with globally lower effective diffusion, which is a faster traveling blob that demonstrates the formation of more fine-scale features in the flow. Finally the third row shows the low diffusion solution without any regularization. Here the snapshots are at $T \approx 0, 2.6, 2.8$ corresponding to timesteps 0, 5000, 10000. It is clear that sharp gradients start to form early in the calculation, leading to sharp numerical instabilities along the leading edges, and jump discontinuities that rapidly pollute the solution and send the CFL condition towards machine epsilon.

A more difficult problem is the case of turbulent edge modes at high resolution. To demonstrate the numerical behavior of ArcOn under such conditions, we take (2.1)-(2.3) with the addition of non-vanishing parallel loss at the wall $\alpha_0 \neq 0$, and a plasma sourcing

$$\eta = \begin{cases} |\sin(y/10)|/100 & \text{if } x \leq 0, \\ 0 & \text{otherwise,} \end{cases}$$

transforming (2.1) into

$$\partial_t n + [\phi, n] = -\alpha_0 n + \eta + D\nabla_{\perp}^2 n.$$

Here the initial state of the system is initialized with,

$$n_0 = e^{-(x^2+y^2)/100}/100 \quad \text{and} \quad \phi_0 = U_0 = 0,$$

effectively producing a density cascade into the dipole center (i.e. the central vanishing isocontour of ϕ) at $y = 0$ as the streamers develop.

This case demonstrates one of the basic difficulties that can emerge in turbulence studies, where large dynamic gradients form in the solution and turbulent vortices cascade into each other at high frequencies. We show this basic result in figure 5. Here 32768 finite elements with discontinuous tensor product of Lagrange polynomials and arbitrary nodal support points are used with $p = 3$ elements solved over 256 processors. The diffusion parameters are set to $D = \mu = 0.05$ and the strong flux forms are used.

In figure 5 we push with a sourcing that is quite strong, as $\eta \sim 100\alpha_0$ leading to plasma accumulation along the edge. Here we have used the alternative form of the equations as in [26], where $\beta = 5\alpha_0 n$

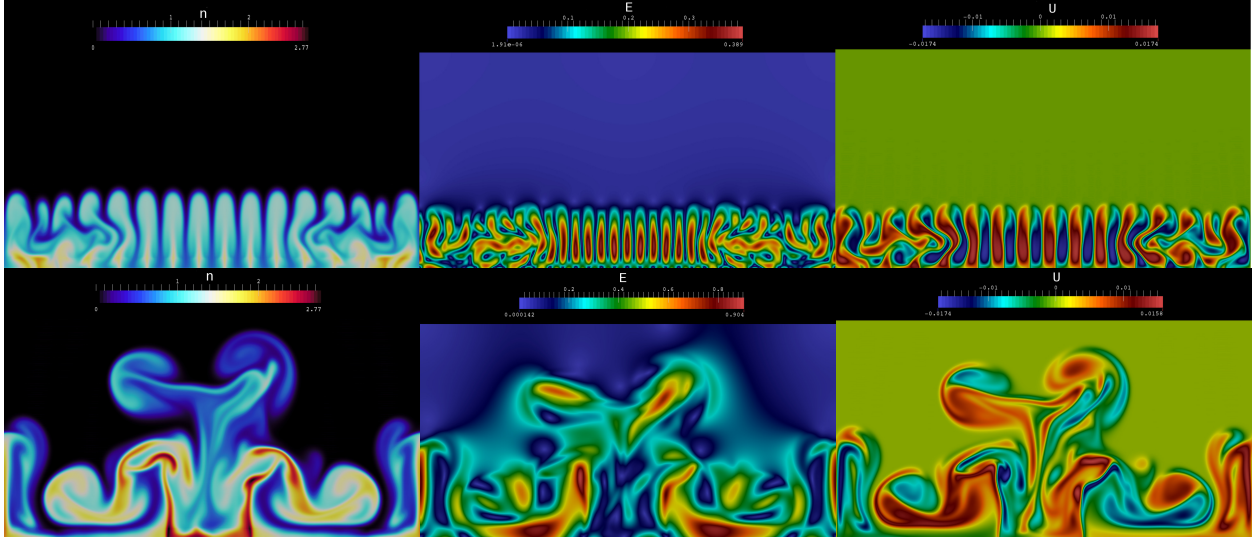


Figure 5: The top graphs show n , \mathbf{E}_\perp and U at $T = 1133.8$, while the bottom shows the same components at $T = 1979.6$.

and $\alpha = 10^{-4}$. In this case we are able to recover the solution using the strong flux, and as is clear in figure 5, sharp profiles along edges are being nicely captured, even without the addition of artificial viscosity.

§6 Physical solutions

6.1 Modon solutions

Physical modons (or dipole drift vortices) emerge as important solutions in a number of fields, such as geophysical fluid dynamics [42], coastal dynamics [65] and atmospheric [27, 57]. Intuitively these objects are often discussed in tandem with their counterpart, monopolar vortices, which in the parlance of atmospheric science correspond to phenomena such as cyclones. In plasma physics applications, monopolar and dipolar drift vortices (modons) [49] are often described as the coherent structures that serve as the fundamental constituents comprising strong drift-wave turbulence. Plasma modons have been used extensively to develop an understanding of magnetized [33] and electrostatic [70] plasmas.

Here we use a modon solution for (2.1)–(2.3) as developed in [49]. The basic features of the solution are such that in the vanishing diffusion/viscosity limit the plasma density becomes a linear function of the electrostatic potential $n(\phi)$, characterized by,

$$n = (\lambda/c)(\phi - cx),$$

where λ and c are parameters.

The potential ϕ is a composite $C^2(\Omega)$ function (see figure 6), that takes the form

$$\phi(r, \theta) = \begin{cases} AK_1(\kappa r/a) \cos \theta & \text{for } r > a \\ (Br/a) \cos \theta + CJ_1(\gamma r/a) \cos \theta & \text{for } r \leq a \end{cases} \quad (6.1)$$

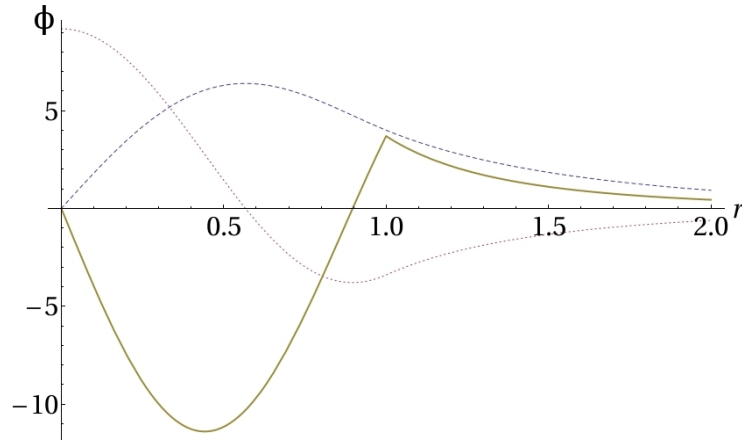


Figure 6: Here we show the $C^2(\Omega)$ function ϕ and its derivatives as a function of r . In dashed blue is ϕ , dotted magenta is $\phi' = \partial_r \phi$, and solid yellow is $\phi'' = \partial_{rr} \phi$.

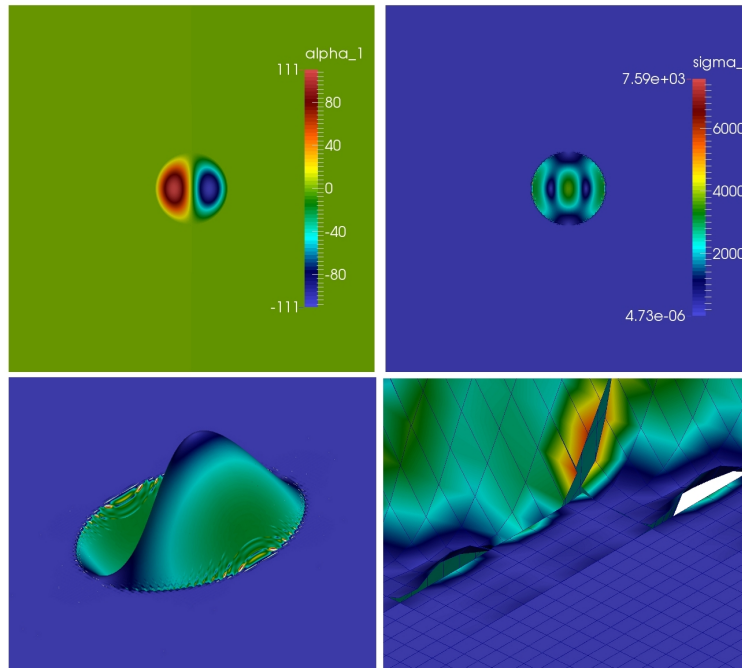


Figure 7: On the top left the electrostatic potential $\alpha_1 = \phi$ of the modon solution and the top right the modulus of its gradient $\sigma_1 = |\nabla_{\perp} U|$, shown after the initial interpolation. The bottom left is a closeup of the mesh along the halo $r = a$ for σ_1 , and the bottom right is a closer zoom-in showing the presence of Gibbs phenomena.

where J_1 and K_1 are Bessel functions of the first and second kind, respectively, $r = \sqrt{x^2 + y^2}$, $\kappa = a\sqrt{\lambda\beta}/c$, $A = ca/K_1(\kappa)$, $B = ca(1 + \kappa^2/\gamma^2)$, $C = (\kappa^2 ca)/(\gamma^2 J_1(\gamma))$, and γ is determined by solving

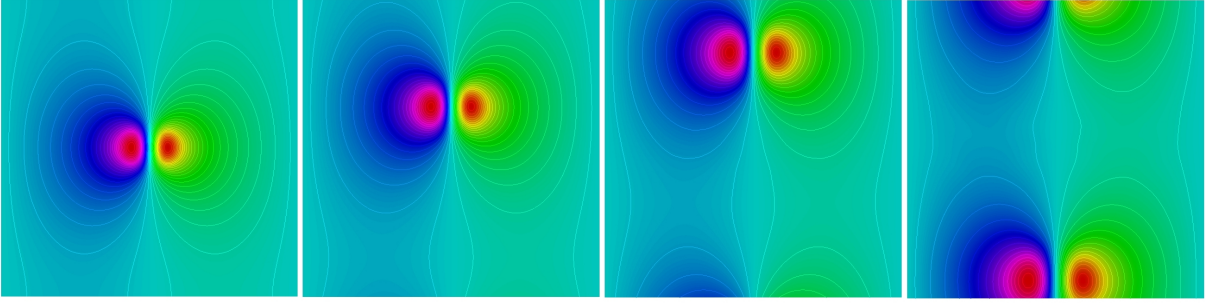


Figure 8: The electrostatic potential ϕ of the modon solution, shown at $t = 0, 0.16, 0.45, 0.97$. The artificial diffusion parameters for the potential field ϕ are $S_0 = 77$ and $\kappa = 80$.

the roots of

$$\frac{K_2(\kappa)}{\kappa K_1(\kappa)} = -\frac{J_2(\gamma)}{\gamma J_1(\gamma)}.$$

The modon solution here poses a challenging numerical problem, as somewhat analogous to [29], the composite function that comprises the potential, plasma density, and vorticity, does not demonstrate enough regularity to smoothly project/interpolate into the expected solution space. Namely here, the continuity equation provides that $\partial_t \phi = c \partial_y \phi$, so that the momentum equation is characterized by third order time-independent spatial differential operators. However, $\phi \in C^2$ prescribes no higher order continuity in the field by construction. The initial state of the system must be constructed first from computing the Laplacian of ϕ to recover the vorticity. This means that $\nabla_{\perp} U$, which drives the convective transport in the flux, demonstrates a discontinuity in the initial state (e.g. the third spatial derivative of a C^2 function) that not surprisingly generates numerical Gibb’s oscillations in the initial data that can be seen to high spatial refinement (see again figure 6, where the interpolation is shown in figure 7 for $p = 4$ and $h = 1.52 \times 10^{-5}$).

Normally such solutions pose substantial problems for DG solution systems, as given an unstable initial state, the high order accuracy/basis of DG schemes tend to amplify singular irregularities, as they comprise inherent features of the mathematical formulation that drive the system. However, it is also the case that physical models of composite systems are often formulated to obey C^j continuity for j smaller than expected as a formal necessity of developing meaningful predictive models. That is, many of the discontinuities that present themselves in these formulations are “nonphysical” in that they are but artifacts of the system numerics that do not emerge as compelling physical features in the validation of the laboratory experiments.

As such, the artificial diffusion from section 5 serves as a panacea of sorts for coupling to systems characterized by these particular types of mathematical pliability. The modon is a perfect example of such a case. Here, with the addition of a weak and locally sharp (or confined) artificial diffusion/viscosity, a stable solution is obtained indefinitely in $(0, T]$. The results are shown in figure 8. The problem is run with periodic boundaries in y , radiation boundaries in x , and the strong form of the numerical fluxes are used. Notice here that the modon itself is not a periodic solution, as the electrostatic potential distorts the electric field once the leading edge is close enough to the periodic boundary to impact the tail of the solution from the other side (i.e. this modon solution “runs into itself”). Also note that up to $t = 0.1185$ the average modon velocity is 1.0015 ± 0.0174 , such that the modon has traveled ~ 12 of the domain (this is slightly before the second snapshot in figure 8), which

is larger than the characteristic modon height and right before the leading edges of ϕ pass across the periodic boundary; hence matching the analytic solution c until this point, as expected. After the leading edge passes across the boundary and thereafter, for example at $T = 0.7267$, the speed averages to 0.8531 ± 0.0405 as the effects from the non-periodicity (self-interaction) cause a persistent slowdown. The parameters in these runs were, $\Omega = [0, 1]^2$, $\beta = a = c = 10\lambda = 1$.

The most remarkable feature of the solution adjoined to the artificial diffusion/viscosity is the effect on the instantaneous wavespeed. The initial interpolation of the base fields into the basis leads to an approximate computed local maximum wavespeed of 1.4×10^4 (this is of course due to the discontinuity in U across edges). The CFL restriction on the explicit timestepper is clearly restrictive in this setting. However, with the modest artificial diffusion field notated in figure 8, we are able to run the variable timestepper with the CFL constraint one hundreds times smaller than prescribed, and still visually maintains a stable solution unpolluted with trace oscillations. This is appealing behavior, and in contrast to a pseudospectral solution behavior in a discontinuous setting for example, where the oscillations caused by Gibbs phenomena may become so remarkably stable (e.g. “resonant” in periodic domains) that oscillations in the residual become indistinguishable from the physics.

6.2 Turbulence

In physical turbulence observed near the SOL edge in tokamaks [5, 50], the parallel loss terms α can have extremely important affects on the dynamics. In addition the form of the system with β a linear function of n , though frequently convenient and of use in physical studies [26], can suppress nonlinearities due to the toroidal curvature. However, when β scales as a constant we can rescale the system to a stable formulation that demonstrates no difficulties with the vacuum.

To do this, define a new variable χ such that $n = e^\chi$. Multiplying the continuity equation by n^{-1} , (2.1)-(2.2) can be rewritten as

$$\partial_t \chi + [\phi, \chi] = -\alpha_0 + D\nabla_\perp^2 \chi + D(\nabla_\perp \chi)^2, \quad \chi|_{t=0} = n_0, \quad \chi = \chi_b \text{ on } \partial\Omega, \quad (6.2)$$

$$\partial_t U + [\phi, U] = \alpha\phi - \beta[x, \chi] + \mu\nabla_\perp^2 U, \quad U|_{t=0} = U_0, \quad U = U_b \text{ on } \partial\Omega. \quad (6.3)$$

This can easily be recast into the weak form, with $\mathcal{O} = \nabla_\perp \chi$. The only major difference here is the second diffusion term $D(\nabla_\perp \chi)^2$, which in the weak form can be treated simply as the interior term $D\zeta_1 \mathcal{O}^2$.

In this form we have confirmed the qualitative dynamics of the system discussed in detail in [5, 50], where we have shown that the turbulent blobs conform to the inertial length scale. That is, taking the initial condition above with $\beta/\alpha_0 = 5$, and setting $\alpha_0 = 10^{-4}$ we confirm the inertial length scale of $(\beta/\alpha^2)^{1/5}$, which is ~ 8.7 .

We also test the e -folding length λ of this system at turbulent saturation as discussed in [5, 50]. These simulations were run with the same settings as above, but here we set $p = 3$ on 1024 finite elements. Frequently e -folding simulations are run on small domains with absorbing boundary conditions, but to avoid the potential first order boundary error that can arise in dispersive systems (see [43, 53, 54, 74] for more details on this) we instead implement the same problem on a periodic domain $(x, y) \in [0, 1600] \times [0, 377]$, where a y -independent plasma source $\eta(\alpha_0) = 3\alpha_0 e^{-(x-75)^2/200}$ is added to the right hand side of (6.2). The resulting system is run until $T = 50,000$ where saturation conditions are plainly evident.

As we can see at time $\sim 38,000$ in figure 10, the χ and n fields show substantial small-scale turbulent features. To validate and benchmark these results, the e -folding length λ is determined by solving a

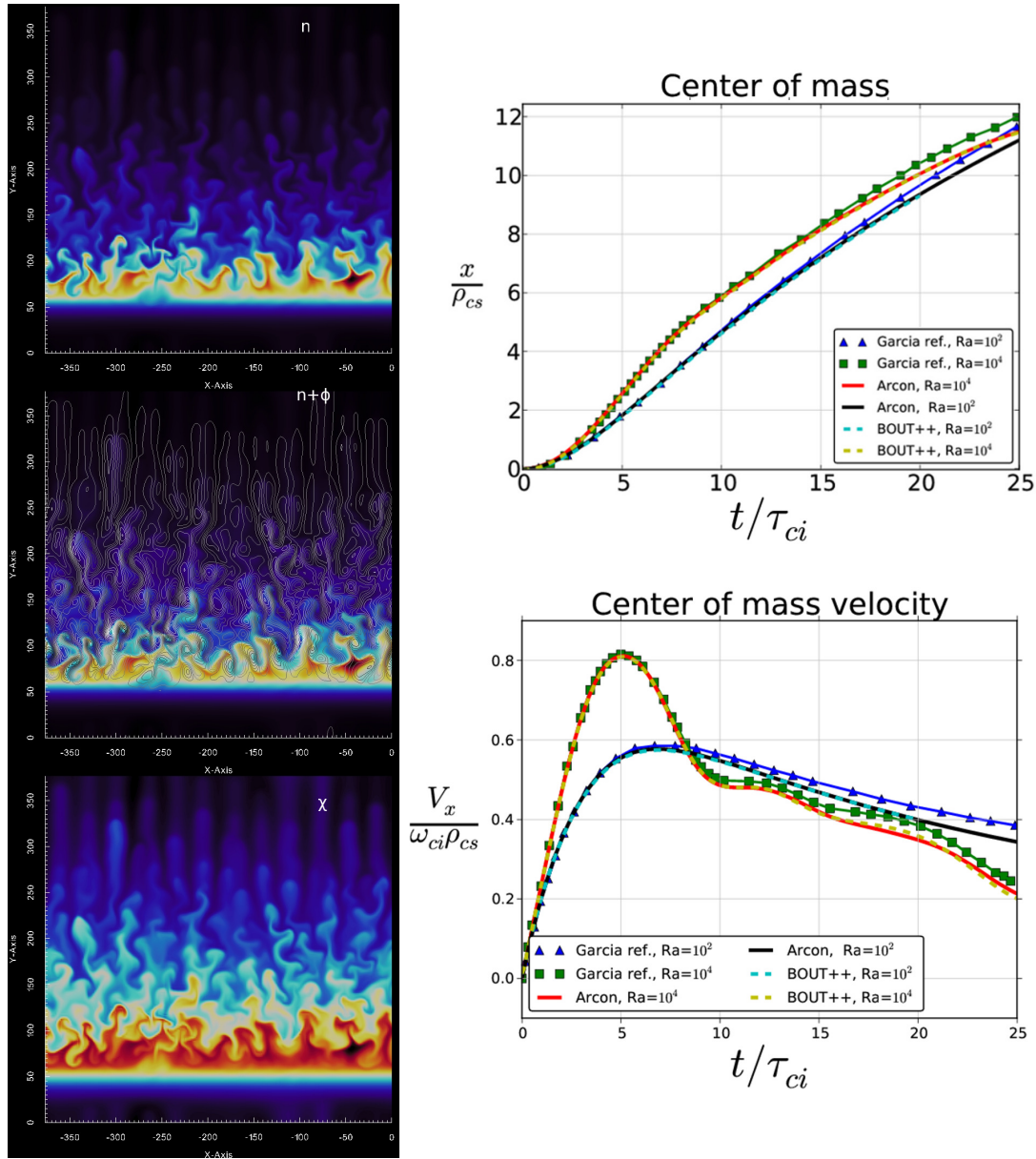


Figure 9: The center of mass and center of mass velocity plots on the right show single blob validation against BOUT++ [50] and Garcia [26] et al. To the left is a turbulent saturation test at $T \sim 38,000$, where the top graph shows the density n , the middle graph shows the density field with ϕ contours overlaid, and the bottom its associated χ field.

least squares fit of $L_y^{-1} \int_y n dy = e^{-x/\lambda}$ for λ , where $L_y = 377$, and the Rayleigh number $Ra = 10^{-4}$. These results are compared to an exact simulation of BOUT++ [50] to find a nearly exact e -folding length of $\lambda = 43$ between the two codes.

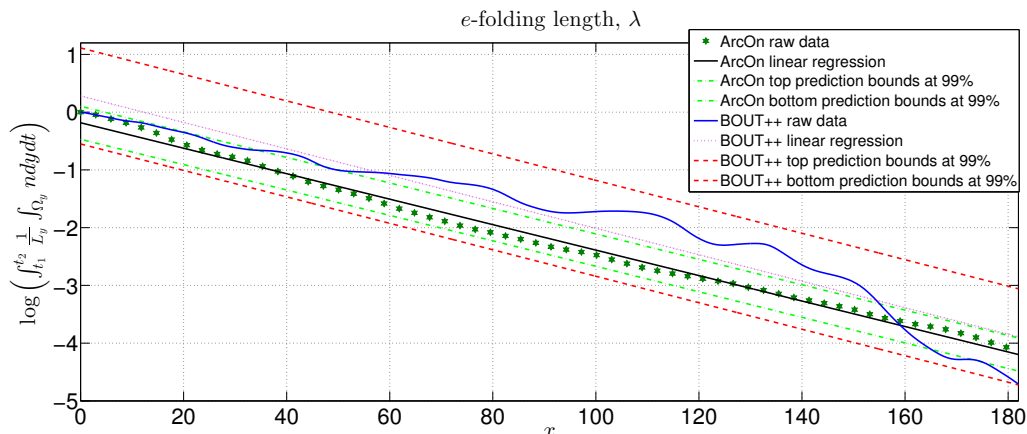


Figure 10: Here we show the equivalent slope and the prediction bound overlap between BOUT++ and ArcOn at turbulent saturation relative to the e -folding length.

§7 Conclusion

We have presented a new architecture called ArcOn for studying the dynamics of filamentary blobs transported through the scrape-off layer of tokamaks. In this regard we have implemented a fully discontinuous Galerkin method for solving (2.1)-(2.3). In contrast to mixed form finite element methods, for example, which use continuous Galerkin projections [53] to recover solutions to Poisson, our present approach preserves the discontinuous function spaces throughout the computation, thus expanding the well-posedness of the space of admissible solutions. Our formulation is novel, in that it supports multiple basis functions, and is run using modal, nodal, or mixed nodal/modal finite elements. In addition, we have developed and analyzed three classes of upwinding schemes, which we discern here as the strong, semistrong, and quasistrong flux formulations developed to handle the particular cross-product driven fluxes that arise in (2.1)-(2.2). We further implement a Poisson solver using a unified implementation that supports, SIPG, NIPG, IIPG, LDG, Brezzi, and Bassi-Rebay fluxes. Future work in this direction include adding functionality to support LDG, LDG2, HDG and Baumann-Oden formulations [9], as well as developing a strongly coupled Riemann solver for the bracketed convective fluxes.

We have carried out a program of thorough numerical verification of the code, which is seen to converge to the prescribed order in each of its respective subsystems, providing numerically smooth initial data. We then discussed some of the regularization aspects of DG methods that render them remarkably flexible. Along these lines we have developed a novel artificial diffusion/viscosity algorithm predicated on a heuristic entropy measure of the relative local variation in the base variable. This regularity sensor S_{ij} is shown in a number of examples to produce robust solutions.

We have validated ArcOn on a collection of physical test cases. The evolution of modon solutions to (2.1)-(2.3), which introduced a number of numerical difficulties (singular behavior in the initial state of the system) are easily managed with modest ε_i . ArcOn was then validated against blob dynamics and benchmarked against BOUT++ [24] and Garcia [26] et al. Finally turbulent saturation behavior was benchmarked against BOUT++ [24]. The direction of future work in these areas include the expansion of the physics in our model to include advanced turbulent features of the flow, such as the effects of electron temperature gradients and magnetic chaos.

§8 Acknowledgements

The first author would like to thank Phillip Schmitz, John Evans, Jed Brown, Timo Heister, Ammar Hakim, Paul Bauman, Georg Stadler, Logan Moon, Kyle Mandli, Andy Terrel, Clint Dawson, and Irene Gamba for fun, insightful, and incredibly helpful discussions. This work was supported by the U.S. Department of Energy under Contract No. DE-FG02-04ER-54742.

References

- [1] C. Altmann, A. Beck, A. Birkefeld, F. Hindenlang, M. Staudenmaier, G. Gassner, and C.-D. Munz. Discontinuous Galerkin for high performance computational fluid dynamics (hp_{cdg}). In W. E. Nagel, D. B. Kröner, and M. M. Resch, editors, *High Performance Computing in Science and Engineering '11*, pages 277–288. Springer Berlin Heidelberg, 2012.
- [2] D. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Discontinuous Galerkin methods for elliptic problems. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 89–101. Springer, Berlin, 2000.
- [3] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.
- [4] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02.
- [5] A. Aydemir. Convective transport in the scrape-off layer of tokamaks. *Physics of Plasmas*, 12(6), Jun 2005.
- [6] S. Balay, J. Brown, , K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.
- [7] W. Bangerth, T. Heister, G. Kanschat, et al. deal.II *Differential Equations Analysis Library, Technical Reference*. <http://www.dealii.org>.
- [8] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997.
- [9] C. Baumann and J. Oden. A discontinuous *hp* finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175(3-4):311–341, JUL 2 1999.
- [10] J. Borggaard, T. Iliescu, and J. P. Roop. A Bounded Artificial Viscosity Large Eddy Simulation Model. *SIAM Journal on Numerical Analysis*, 47(1):622–645, 2009.
- [11] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous finite elements for diffusion problems. in *Atti Convegno in onore di F. Brioschi (Milan, 1997)*, Istituto Lombardo, Accademia di Scienze e Lettere, Milan, Italy, pages 197–217, 1999.

-
- [12] C. Burstedde, L. C. Wilcox, and O. Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.
- [13] Y. Cheng, I. M. Gamba, and P. J. Morrison. Study of conservation and recurrence of Runge-Kutta discontinuous Galerkin schemes for Vlasov-Poisson systems. *Journal of Scientific Computing*, 56(2):319–349, AUG 2013.
- [14] Y. Cheng, F. Li, J. Qiu, and L. Xu. Positivity-preserving DG and central DG methods for ideal MHD equations. *Journal of Computational Physics*, 238:255–280, Apr 1 2013.
- [15] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems. *J. Comput. Phys.*, 84(1):90–113, 1989.
- [16] B. Cockburn and C.-W. Shu. The runge-kutta discontinuous galerkin method for conservation laws v: Multidimensional systems. *Journal of Computational Physics*, 141(2):199 – 224, 1998.
- [17] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2001.
- [18] C. Dawson, S. Sun, and M. F. Wheeler. Compatible algorithms for coupled flow and transport. *Comput. Methods Appl. Mech. Engrg.*, 193(23-26):2565–2580, 2004.
- [19] C. Dawson, C. J. Trahan, E. J. Kubatko, and J. J. Westerink. A parallel local timestepping Runge-Kutta discontinuous Galerkin method with applications to coastal ocean modeling. *Computer Methods in Applied Mechanics and Engineering*, 259:154–165, Jun 1 2013.
- [20] C. Dawson, M. Wheeler, and J. Proft. Adaptive stencil and discontinuous Galerkin methods for transport equations on triangular grids. *Technical Report*, 1999.
- [21] A. Dedner and M. Ohlberger. A new *hp*-adaptive DG scheme for conservation laws based on error control. In *Hyperbolic problems: theory, numerics, applications*, pages 187–198. Springer, Berlin, 2008.
- [22] D. A. D’Ippolito, J. R. Myra, and S. J. Zweben. Convective transport by intermittent blob-filaments: Comparison of theory and experiment. *Physics of Plasmas*, 18(6):060501, 2011.
- [23] B. D. Dudson, M. V. Umansky, X. Q. Xu, P. B. Snyder, and H. R. Wilson. BOUT++: A framework for parallel plasma fluid simulations. *Computer Physics Communications*, 180(9):1467–1480, SEP 2009.
- [24] B. D. Dudson, M. V. Umansky, X. Q. Xu, P. B. Snyder, and H. R. Wilson. BOUT++: A framework for parallel plasma fluid simulations. *COMPUTER PHYSICS COMMUNICATIONS*, 180(9):1467–1480, SEP 2009.
- [25] M. Feistauer and V. Kučera. On a robust discontinuous Galerkin technique for the solution of compressible flow. *J. Comput. Phys.*, 224(1):208–221, 2007.
- [26] O. E. Garcia, N. H. Bian, and W. Fundamenski. Radial interchange motions of plasma filaments. *Physics of Plasmas*, 13(8), AUG 2006.

- [27] M. Ghil, M. Kimoto, and J. Neelin. Nonlinear dynamics and predictability in the atmospheric sciences. *Reviews of Geophysics*, 29(1, S):46–55, 1991.
- [28] R. E. Heath, I. M. Gamba, P. J. Morrison, and C. Michler. A discontinuous Galerkin method for the Vlasov-Poisson system. *Journal of Computational Physics*, 231(4):1140–1174, Feb 20 2012.
- [29] J. Hesthaven, J. Lynov, and J. Nycander. Dynamics of nonstationary dipole vortices. *Physics of Fluids a-Fluid Dynamics*, 5(3):622–629, MAR 1993.
- [30] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. Algorithms, analysis, and applications.
- [31] G. Jacobs and J. Hesthaven. High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *Journal of Computational Physics*, 214(1):96–121, May 1 2006.
- [32] G. B. Jacobs and J. S. Hesthaven. Implicit-explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning. *Computer Physics Communications*, 180(10):1760–1767, OCT 2009.
- [33] D. Jovanovic and P. K. Shukla. Structures and Zonal Flows in Magnetized Plasmas. In Eliasson, B and Shukla, PK, editor, *New Frontiers in Advanced Plasma Physics*, volume 1306 of *AIP Conference Proceedings*, pages 150–159. Abdus Salam Int Ctr Theoret Phys, 2010. ICTP International Advanced Workshop on the Frontiers of Plasma Physics, Abdus Salam Int Ctr Theoret Phys, Trieste, Italy, Jul 05-16, 2010.
- [34] A. Kloeckner, T. Warburton, J. Bridge, and J. S. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228(21):7863–7882, Nov 2009.
- [35] S. I. Krasheninnikov, D. A. D’Ippolito, and J. R. Myra. Recent theoretical progress in understanding coherent structures in edge and SOL turbulence. *Journal of Plasma Physics*, 74(5):679–717, OCT 2008.
- [36] E. Kubatko, C. Dawson, and J. Westerink. Time step restrictions for Runge-Kutta discontinuous Galerkin methods on triangular grids. *J. Comput. Phys.*, 227(23):9697–9710, 2008.
- [37] E. J. Kubatko, J. J. Westerink, and C. Dawson. hp discontinuous Galerkin methods for advection dominated problems in shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):437–451, 2006.
- [38] E. J. Kubatko, J. J. Westerink, and C. Dawson. Semi discrete discontinuous galerkin methods and stage-exceeding-order, strong-stability-preserving runge-kutta time discretizations. *J. Comput. Phys.*, 222(2):832–848, Mar. 2007.
- [39] D. Kuzmin. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *J. Comput. Appl. Math.*, 233(12):3077–3085, 2010.
- [40] D. Kuzmin, M. J. Shashkov, and D. Svyatskiy. A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems. *Journal of Computational Physics*, 228(9):3448–3463, May 20 2009.

-
- [41] D. Kuzmin and S. Turek. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *Journal of Computational Physics*, 198(1):131–158, Jul 20 2004.
- [42] V. D. Larichev and G. M. Reznik. Two-dimensional solitary rossby waves. *Dokl. Akad. Nauk SSSR*, 231:1077, 1976.
- [43] I. Lie. Well-posed transparent boundary conditions for the shallow water equations. *Appl. Numer. Math.*, 38(4):445–474, 2001.
- [44] A. Loarte, B. Lipschultz, A. Kukushkin, G. Matthews, P. Stangeby, N. Asakura, G. Counsell, G. Federici, A. Kallenbach, K. Krieger, A. Mahdavi, V. Philipps, D. Reiter, J. Roth, J. Strachan, D. Whyte, R. Doerner, T. Eich, W. Fundamenski, A. Herrmann, M. Fenstermacher, P. Ghendrih, M. Groth, A. Kirschner, S. Konoshima, B. LaBombard, P. Lang, A. Leonard, P. Monier-Garbet, R. Neu, H. Pacher, B. Pegourie, R. Pitts, S. Takamura, J. Terry, E. Tsitrone, the ITPA Scrape-off Layer, and D. P. T. Group. Chapter 4: Power and particle control. *Nuclear Fusion*, 47(6):S203–S263, 2007.
- [45] J. Loverich, A. Hakim, and U. Shumlak. A Discontinuous Galerkin Method for Ideal Two-Fluid Plasma Equations. *Communications in Computational Physics*, 9(2):240–268, Feb 2011.
- [46] J. Loverich and U. Shumlak. A discontinuous Galerkin method for the full two-fluid plasma model. *Computer Physics Communications*, 169(1-3):251–255, Jul 1 2005. Europhysics Conference on Computational Physics, Genoa, ITALY, SEP 01-04, 2004.
- [47] A. Mani, J. Larsson, and P. Moin. Suitability of artificial bulk viscosity for large-eddy simulation of turbulent flows with shocks. *Journal of Computational Physics*, 228(19):7368–7374, OCT 20 2009.
- [48] E. T. Meier, A. H. Glasser, V. S. Lukin, and U. Shumlak. Modeling open boundaries in dissipative MHD simulation. *Journal of Computational Physics*, 231(7):2963–2976, Apr 1 2012.
- [49] J. Meiss and W. Horton. Solitary vortices in the presence of magnetic shear. *Phys. Plasmas*, 26(4):990–997, April 1983.
- [50] D. Meyerson, C. Michoski, W. Horton, and F. Waelbroeck. Radial interchange driven turbulence and transport across chaotic magnetic field lines. *In preparation*, 2014.
- [51] C. Michoski, A. Alexanderian, C. Dawson, and E. Kubatko. Numerical stability analysis for explicit runge-kutta methods in nonlinear discontinuous systems. *preprint*, 2014.
- [52] C. Michoski, J. Evans, and P. Schmitz. Discontinuous Galerkin *hp*-adaptive Methods for Multi-scale Chemical Reactors I: Quiescent Reactors. *preprint*, 2014.
- [53] C. Michoski, J. A. Evans, P. G. Schmitz, and A. Vasseur. Quantum hydrodynamics with trajectories: the nonlinear conservation form mixed/discontinuous Galerkin method with applications in chemistry. *J. Comput. Phys.*, 228(23):8589–8608, 2009.
- [54] C. Michoski, J. A. Evans, P. G. Schmitz, and A. Vasseur. A discontinuous Galerkin method for viscous compressible multifluids. *J. Comput. Phys.*, 229(6):2249–2266, 2010.

- [55] C. Michoski, C. Mirabito, C. Dawson, E. Kubatko, D. Wirasaet, and J. Westerlink. Adaptive hierarchic transformations over dynamic p -enriched schemes applied to generalized DG systems. *J. Comput. Phys.*, *submitted*, 2010.
- [56] P. Morrison. A paradigm for joined hamiltonian and dissipative systems. *Physica D*, 18(1-3):410–419, Jan 1986.
- [57] E. Neven. Linear stability of modons on a sphere. *Journal of the Atmospheric Sciences*, 58(16):2280–2305, 2001.
- [58] J. T. Oden, I. Babuška, and C. E. Baumann. A discontinuous hp finite element method for diffusion problems. *J. Comput. Phys.*, 146(2):491–519, 1998.
- [59] J. L. Peterson and G. W. Hammett. Positivity preservation and advection algorithms with applications to edge plasma turbulence. *SIAM Journal on Scientific Computing*, 35(3):B576–B605, 2013.
- [60] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations*, volume 35 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Theory and implementation.
- [61] J. A. Rossmannith and D. C. Seal. A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations. *Journal of Computational Physics*, 230(16):6203–6232, Jul 10 2011.
- [62] D. A. Russell, J. R. Myra, and D. A. D’Ippolito. Saturation mechanisms for edge turbulence. *Physics of Plasmas*, 16(12):122304, 2009.
- [63] S. Ruuth. Global optimization of explicit strong-stability-preserving Runge-Kutta methods. *Math. Comp.*, 75(253):183–207 (electronic), 2006.
- [64] S. K. Saha and S. Chowdhury. Intermittent transport in the scrape-off layer of the SINP tokamak. *Physics of Plasmas*, 13(9), SEP 2006.
- [65] H. Sakuma and M. Ghil. Stability of propagating modons for small-amplitude perturbations. *Physics of Fluids a-Fluid Dynamics*, 3(3):408–414, MAR 1991.
- [66] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.
- [67] U. Shumlak, R. Lilly, N. Reddell, E. Sousa, and B. Srinivasan. Advanced physics calculations using a multi-fluid plasma model. *Computer Physics Communications*, 182(9, SI):1767–1770, SEP 2011.
- [68] B. Srinivasan, A. Hakim, and U. Shumlak. Numerical Methods for Two-Fluid Dispersive Fast MHD Phenomena. *Communications in Computational Physics*, 10(1):183–215, Jul 2011.
- [69] T. Toulorge and W. Desmet. {CFL} conditions for runge-kutta discontinuous galerkin methods on triangular grids. *Journal of Computational Physics*, 230(12):4657 – 4678, 2011.

-
- [70] F. Waelbroeck, P. Morrison, and W. Horton. Hamiltonian formulation and coherent structures in electrostatic turbulence. *Plasma Physics and Controlled Fusion*, 46(9):1331, 2004.
- [71] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, Jul 20 2013.
- [72] T. Warburton and G. Karniadakis. A discontinuous Galerkin method for the viscous MHD equations. *Journal of Computational Physics*, 152(2):608–641, Jul 1 1999.
- [73] L. Wilcox, G. Stadler, C. Burstedde, and O. Ghattas. A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media. *Journal of Computational Physics*, 229(24):9373–9396, DEC 10 2010.
- [74] D. Wirasaet, S. Brus, C. Michoski, E. Kubatko, J. Westerink, and C. Dawson. Artificial boundary layers in the discontinuous Galerkin shallow water solution in channels. *preprint*, 2013.
- [75] D. Wirasaet, S. Tanaka, E. J. Kubatko, J. J. Westerink, and C. Dawson. A performance comparison of nodal discontinuous Galerkin methods on triangles and quadrilaterals. *International Journal for Numerical Methods in Fluids*, 64(10-12):1336–1362, Dec 30 2010. 15th International Conference on Finite Elements in Flow Problems, Tokyo, Japan, Apr 01-03, 2009.
- [76] V. Zingan, J.-L. Guermond, J. Morel, and B. Popov. Implementation of the entropy viscosity method with the discontinuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 253:479–490, JAN 1 2013.