UK Atomic Energy Authority

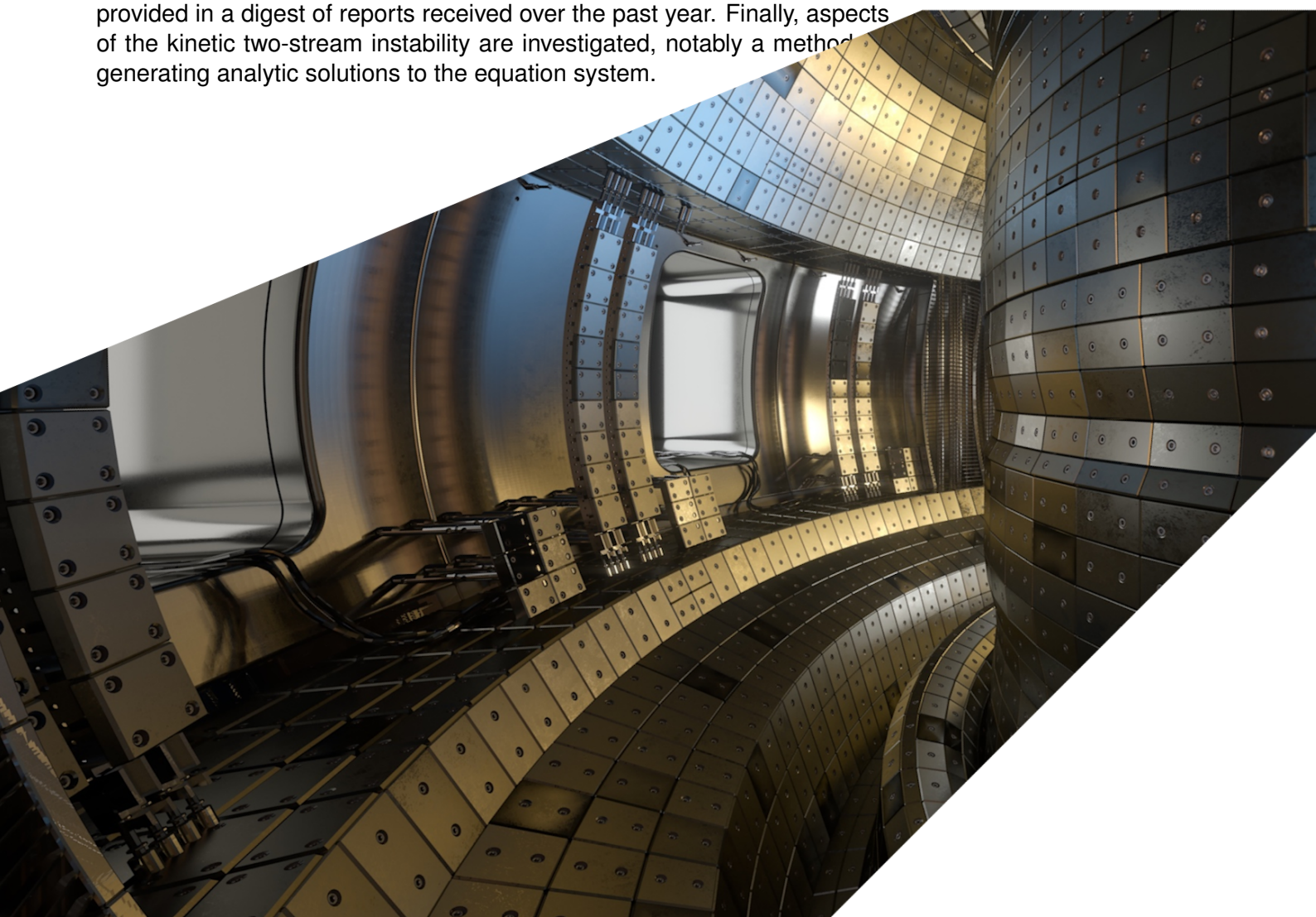
ExCALIBUR

Complementary actions. Code integration, acceptance and operation 3.

M6c.3 Version 1.01

Abstract

The report describes work for ExCALIBUR project NEPTUNE at Milestone M6c.3. It provides material concerning the implementation of the HERMES-3 equations in NEKTAR++ in support of recent work by the holder of the grant T/AW085/22, covering the initial version written by the grant holder and further developments in order to improve the efficiency of the solver; also some attempts to verify the outputs of the aforementioned codes are outlined. In view of the need to couple other physics codes to NEKTAR++, the interface for adding source terms into the NEKTAR++ incompressible Navier-Stokes solver is demonstrated. Additional work outlines an implementation of convection problems within the FIREDRAKE PDE solver framework using a simple continuation method to access flows at higher values of the Rayleigh number; this code is verified against some recent results from the literature. An overview of relevant grantee work is provided in a digest of reports received over the past year. Finally, aspects of the kinetic two-stream instability are investigated, notably a method generating analytic solutions to the equation system.



UKAEA REFERENCE AND APPROVAL SHEET

	Client Reference:	
	UKAEA Reference:	CD/ExCALIBUR -FMS/0074
	Issue:	1.01
	Date:	23 December 2023
Project Name: ExCALIBUR Fusion Modelling System		

	Name and Department	Signature	Date
Prepared By:	Ed Threlfall Owen Parry Wayne Arter CD	N/A N/A N/A	23 March 2023 23 March 2023 23 December 2023
Reviewed By:	Wayne Arter CD		23 December 2023

1 Introduction

The development of high order (spectrally) accurate finite element models is seen as critical for the production of efficient Exascale software. Deliverable 6c is designed to enable an uninterrupted programme of development, providing continuity for external work on spectral elements and an advanced referent model.

Section 2 of this report is concerned with a spectral / hp element implementation of a set of plasma fluid equations. Specifically, System 2-6 of the revised NEPTUNE proxy-app equations in [1] is a spatially two-dimensional plasma model incorporating velocity-space effects. This is intended to be simulated using two-dimensional finite elements with kinetic effects incorporated eg. by coupling to a particle representation that acts to introduce source terms into the fluid model (intended to represent eg. ionization of kinetic neutrals). Physically this picture represents a collisional plasma interacting with a kinetic population of particles (eg. fusion-born ions or neutrals sourced from the reactor walls having been created by bombardment of fusion-born ions). Kinetic species' models are also needed for the interaction of the plasma with the first wall of the reactor (the so-called plasma 'sheath' terms), though note that, in the equations described in this report, a much simpler representation of the sheath terms is used. The plasma fluid model described in the current version of [1] mirrors that implemented in the existing HERMES-3 multispecies drift-reduced simulation code. HERMES-3 is based on the BOUT++ framework and it uses a finite-difference spatial discretization; one target of the current work is an exploration of the benefits of an implementation using spectral / hp methods. A discussion of initial attempts to implement a subset of the HERMES-3 capability within NEKTAR++ is followed by a description of some possible validation tests applicable to the implementation. There is also a small amount of material explaining how source terms might be implemented in the NEKTAR++ framework.

The FIREDRAKE PDE solver framework is an alternative to NEKTAR++ and this can provide expedient implementations of relevant physics problems with the flexibility of its user interface a major strength. Section 3 contains an explanation of how to obtain the Nusselt number for laminar solutions to a convecting system, including a comparison with recent results from the literature. A simple continuation method for obtaining solutions for large values of the Rayleigh number is presented.

Section 4 summarizes the various reports from grantees that are relevant to this work stream, with subsections covering reports from King's College London and the University of Oxford.

An appendix contains a small amount of mathematical analysis for a kinetic model (the two-stream instability) and a small study of how this kinetic model behaves differently to a classical fluid. This is of course in keeping with a key theme of NEPTUNE which is the relationship between the two types of model (kinetic and fluid) and how to couple them, hence it is expected that this sort of study will aid in the future coupling of these systems.

2 HERMES-3 equations implementation in NEKTAR++

It has been decided to exchange the plasma fluid equations to the system used in the HERMES-3 multifluid drift reduced model (a particular limit of the physics governing the plasma that is relevant

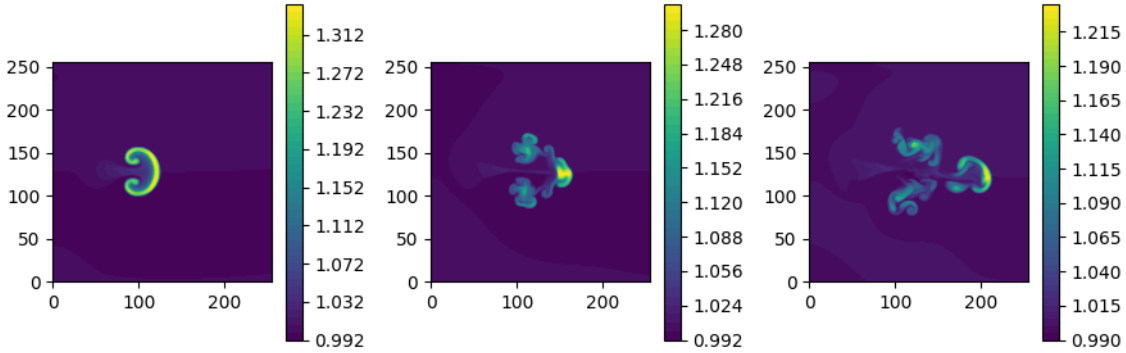


Figure 1: Plots showing the electron density after 15, 30, 45 time steps in the HERMES-3 `blob2d` example.

in realistic tokamak scenarios).

HERMES-3 is an established plasma simulation code with an existing user base, which should mean that a degree of verification is possible. The ultimate goal is a full three-dimensional turbulence simulation code; here, an incremental approach is adopted with development proceeding via the implementation and testing of reduced versions of the equations.

As a starting point, attention will be focussed on the `blob2d` example which can be found at [2]. This is intended to be a two-dimensional model of extended filamentary structures commonly observed in tokamak experiments, referred to as plasma filaments or 'blobs' (one reference is [3]). An important more recent reference on modelling the motion of blobs is [4].

The equations, taken from the HERMES-3 documentation, are

$$\begin{aligned}
\frac{\partial \omega}{\partial t} &= -\nabla\cdot(\omega \mathbf{v}_{E\times B}) + \nabla\left(p_e \nabla\times\frac{\mathbf{b}}{B}\right) + \nabla\cdot\mathbf{j}_{sh};\\
\frac{\partial n_e}{\partial t} &= -\nabla\cdot(n_e\mathbf{v}_{E\times B}) + \frac{1}{e}\nabla\cdot\mathbf{j}_{sh};\\
\nabla\cdot\left(\frac{1}{B^2}\nabla_\perp\phi\right) &= \omega;\\
p_e &= e n_e T_e.
\end{aligned} \tag{1}$$

Here n_e is the electron density, ω is the plasma vorticity, ϕ is the electrostatic potential. The electron temperature T_e is taken to be a constant and p_e is the electron pressure. The strength of the magnetic field perpendicular to the plane is B. The so-called 'E cross B drift' velocity is given by

$$\mathbf{v}_{E\times B} = \frac{\mathbf{E}\times\mathbf{B}}{B^2} \tag{2}$$

where the electric field is purely electrostatic, $\mathbf{E} = -\nabla\phi$.

The x axis in this model corresponds to the radial coordinate in the tokamak (the left is the inboard side) and the (x, y) plane is a slice at constant toroidal angle.

A simple (non-kinetic) sheath closure is used

$$\nabla \cdot \mathbf{j}_{sh} = \frac{n_e \phi}{L_\parallel} \tag{3}$$

in which $L_\parallel$ is the connection length; this is supposed to model the effect of interaction with the material surfaces on which the magnetic field lines end.

Running the basic `blob2d` example and processing the output in the suggested manner generates Fig.1. It can be seen that an initial region of electron density undergoes non-linear evolution to a turbulent state. Note that this example runs in c.5 minutes using one core on a modern laptop.

2.1 Implementation of `blob2d` example in NEKTAR++

2.1.1 Preliminary implementation

An initial implementation followed the same pattern as the *Nektar-Driftwave* example available at [5] in using an explicit time-evolution scheme with no physical dissipative terms. Indeed, once written out explicitly, the equations are somewhat similar to the Hasegawa-Wakatani system used in the aforementioned proxyapp, being

$$\begin{aligned} \frac{\partial n}{\partial t} - \frac{1}{B}\left(\frac{\partial \phi}{\partial x}\frac{\partial n_e}{\partial y} - \frac{\partial \phi}{\partial y}\frac{\partial n_e}{\partial x}\right) &= \frac{n_e \phi}{L}; \\ \frac{\partial \omega}{\partial t} - \frac{1}{B}\left(\frac{\partial \phi}{\partial x}\frac{\partial \omega}{\partial y} - \frac{\partial \phi}{\partial y}\frac{\partial \omega}{\partial x}\right) &= \epsilon \frac{\partial n_e}{\partial y} + \frac{n_e \phi}{L}; \\ \nabla^2 \phi &= B^2 \omega. \end{aligned} \tag{4}$$

The term $\frac{\partial n_e}{\partial y}$ represents the so-called diamagnetic drift and is caused by the curvature of the magnetic field perpendicular to the drift plane (it is absent if the magnetic field is assumed to be uniform). The terms $\frac{n_e \phi}{L}$ called connection terms are representative of the divergence of the sheath current as indicated above (Equation (3)) and are intended to model what happens at the termini of the field lines perpendicular to the drift-plane.

This implementation was developed by the holder of Grant T/AW085/22 in collaboration with UKAEA, and is available at [6]. A simple test involves an initial blob of electron density and this evolves in time under the influence of a force coming from the diamagnetic drift (basically a centrifugal force) and the output is shown in Fig.2. Note that the implementation contains no explicit viscosity or number density diffusivity, and the spectral / hp method means that there is relatively little numerical viscosity or diffusivity. The physics therefore closely approximates that of an ideal (Euler) fluid and this manifests in the generation of structure at all lengthscales, leading the flow to develop a fractal-like appearance. This is not observed in the equivalent output of the HERMES-3 version (Fig.1).

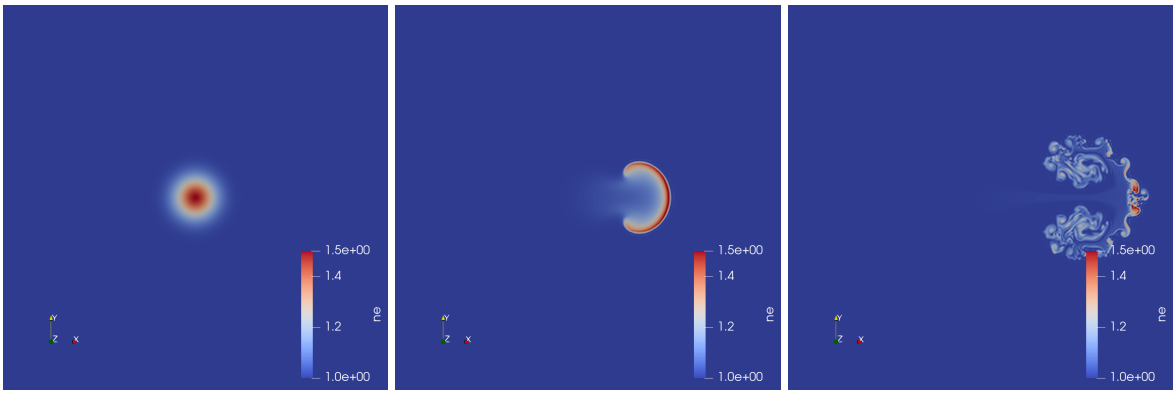


Figure 2: Plots showing the time evolution of the initial NEKTAR++ `blob2d` example.

It would seem natural to add an explicit viscosity term and an explicit diffusivity term into these equations in order to bring the outputs closer to those observed in HERMES-3 (note that at this point it is not particularly clear which of the two types of dissipation - temperature or momentum - is more important). One notable problem with this, however, is that the explicit time-evolution is expected to suffer from a very restricted maximum stable timestep once viscosity or diffusivity is added (this stability problem is present for hyperbolic systems, typically in the form $\frac{|v|\Delta t}{\Delta x} \leq 1$ for maximum velocity $|v|$, but is worse for parabolic systems as there the stability criterion is of the form $\frac{D\Delta t}{(\Delta x)^2} \leq 1$ for diffusivity D). For this reason an alternative approach was considered and this is explained in the sequel.

2.1.2 Implementation as a fluid convection problem - semi-implicit

Given the restriction on the timestep associated to the diffusion terms, it is natural to seek at least a semi-implicit formulation. It is possible to reformulate the system 4 as a convection problem, in which the diamagnetic drift term represents the buoyancy force in the Boussinesq approximation (in this picture, the gravitational field would be horizontal and pulling to the left). This will mean that the existing incompressible Navier-Stokes solver of NEKTAR++ can be used as a starting point.

The Navier-Stokes equations, coupled to thermally-driven buoyancy (assuming also the Boussinesq approximation) acting in the positive x-direction, are

$$\begin{aligned} \frac{\partial \Omega}{\partial t} + v \cdot \nabla \Omega &= \nabla^2 \Omega - \mathrm{Gr}\, \frac{\partial T}{\partial y}; \\ \frac{\partial T}{\partial t} + v \cdot \nabla T &= \frac{1}{\mathrm{Pr}} \nabla^2 T; \\ \nabla^2 \psi &= -\Omega. \end{aligned} \tag{5}$$

where $\Omega = \nabla \times \underline{v}$ is the vorticity, T is the temperature, and ψ is the streamfunction (which implies the incompressibility constraint since $\underline{v} = \nabla \times \psi$). The non-dimensionalization corresponds to Section B.3; the timescale is set by the kinematic viscosity, the quantity Gr is the Grashof number

which measures the strength of the buoyancy force, and the Prandtl number Pr term is the ratio of kinematic viscosity to temperature diffusivity. Note that the vorticity formulation is used here, whereas the Navier-Stokes implementation in NEKTAR++ uses the velocity formulation.

The fluid analogy further proceeds by the identifications

$$\begin{aligned} \Omega &= -\omega; \\ T &= n; \\ \phi &= \psi; \end{aligned} \tag{6}$$

where the quantities on the right hand side are refer to the plasma and those on the left to an ordinary fluid.

The connection terms are not included in this initial work and so the first step is to examine the behaviour of the HERMES-3 equations in which the connection terms are suppressed. This is done by setting the connection length to a large value. It is found that this changes the evolution of the blob - Fig.3. The execution time also increased from c.5 minutes to c.22 minutes with $L_{\|} = 10^4$ replacing the value $L_{\|} = 10$ found in the example. This gives a picture of blob evolution in the absence of the connection terms (ie. a pure Navier-Stokes system) and so outputs of NEKTAR++ can be compared with this.

An initial experiment gave outputs bearing superficial resemblance to the outputs of HERMES-3 `blob2d` with the connection terms suppressed by setting $L_{\|} = 10^4$ - see fig.4. Note that this run took c.20 minutes using one core of a modern laptop. Two values of Pr were tried. The value of Gr was determined empirically ie. to generate something giving a rough resemblance to the HERMES-3 output (the specific value used was $\mathrm{Gr} = 5 \times 10^9$).

The equations can be seen to reproduce some features of the HERMES-3 outputs though the Grashof number appropriate to the numerical viscosity in HERMES-3 must be guessed. This is because there is a grid-dependent numerical viscosity in HERMES-3 (see [7], A.10.2).

Also the effective diffusivity for the temperature is presumably also a grid-dependent numerical effect. In fact, experiments with HERMES-3 show that, at least for time-stationary fields (zero velocity) there is *zero* temperature diffusivity; this is shown by the fact that the blob initial state is time-invariant if the force term `bxcvz` is set to zero (it does not diffuse) (Fig.5) and also a preliminary HERMES-3 implementation of the 'hot layer' example in the following section does not show any diffusion (in fact it is stationary unless some perturbation is seeded into the initial data). Note that zero temperature diffusivity is *not* consistent with the appearance of HERMES-3 outputs (Fig.4) because as Pr is made smaller, the system develops structure at small scales and so one is forced to conclude that there is some sort of velocity-dependent diffusivity acting to smear out the time evolution of the temperature. Note also that the NEKTAR++ incompressible Navier-Stokes solver cannot be run with zero temperature diffusivity, due to the presence of the code line `1.0/aii_Dt/m_diffCoeff[i]` in the source code (`VelocityCorrectionScheme.cpp`) - so zero diffusivity here leads to a divide by zero.

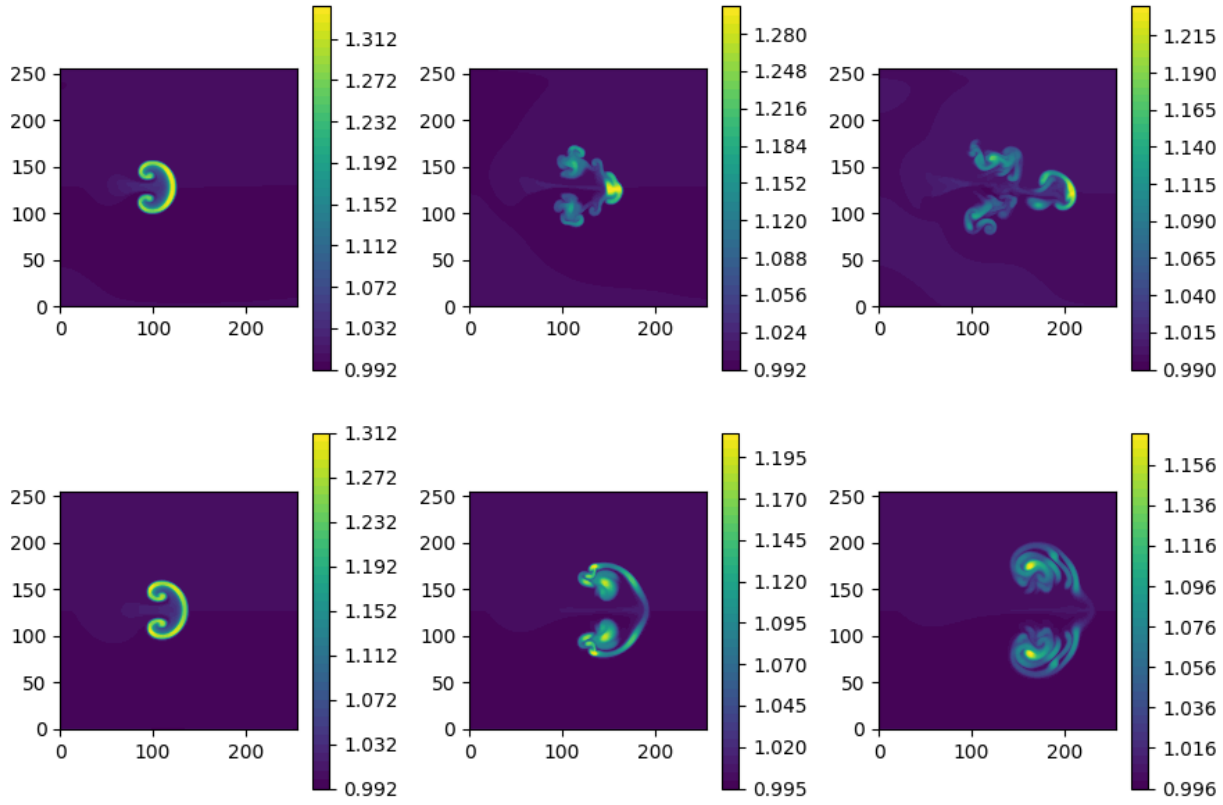


Figure 3: Plots showing the electron density after 15, 30, 45 time steps in the HERMES-3 `blob2d` example. The top sequence has $L_{||} = 10$ (default) and lower one has $L_{||} = 10^4$, the latter suppressing the connection terms.

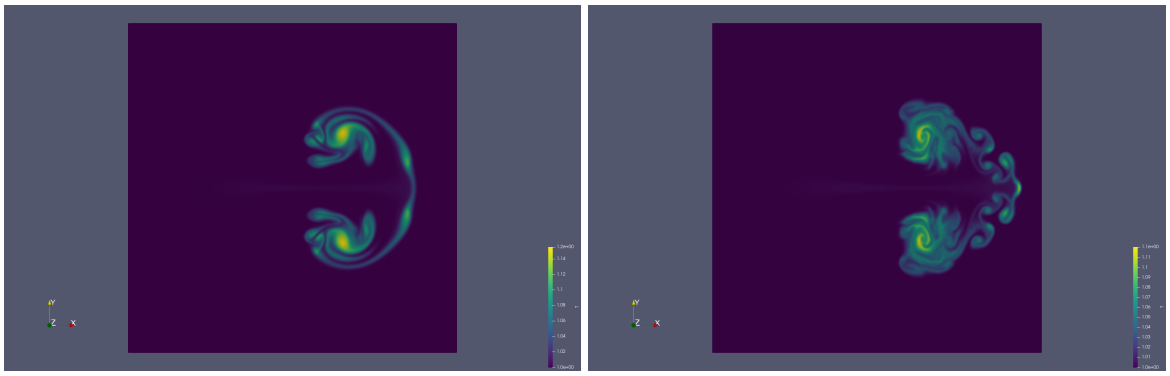


Figure 4: Initial experiments with the NEKTAR++ `blob2d` example (NEKTAR++ temperature field shown). These used $\mathrm{Gr} = 5 \times 10^9$ and $\mathrm{Pr} = 1.0$ (L) and $\mathrm{Pr} = 0.1$ (R).

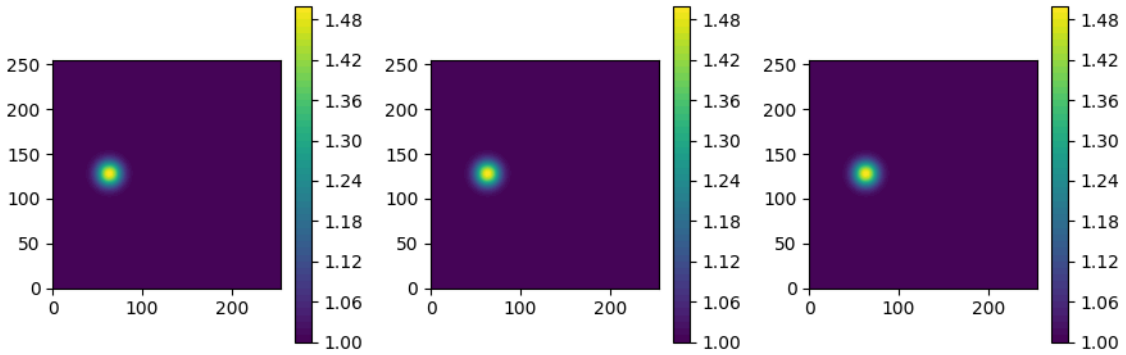


Figure 5: Plots showing the electron density after 15, 30, 45 time steps in the HERMES-3 `blob2d` example with the force term `bxcvz` set to zero (otherwise parameters are as in the original `blob2d` example). It is seen that no diffusion occurs in the initial temperature profile. The same example was also run for 100 times the default duration (by changing the number of timesteps `nout` and amending the output script to grab fields at different times) and again no discernible change was seen in the output.

2.2 Implementation of connection terms

The connection terms are the simplest implementation of the effect of the plasma sheath on which the magnetic field lines terminate. These are the terms Equation (3) in the system Equation (1); the vorticity equation and the density equation are modified by the addition of a forcing term $\frac{n\phi}{L_\parallel}$.

Techniques are required to convert between the velocity formulation of the NEKTAR++ incompressible Navier-Stokes solver and the vorticity formulation of the HERMES-3 equations.

The potential can be found by solving the scalar Poisson problem with the vorticity as the source term. The forcing terms needed for the velocity components u_x and u_y can be derived by solving

$$\frac{\partial u_x}{\partial t} = f_x; \tag{7}$$

$$\frac{\partial u_y}{\partial t} = f_y, \tag{8}$$

so that

$$\frac{\partial \omega}{\partial t} = \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \tag{9}$$

then if

$$(f_x, f_y) = \nabla \times g \tag{10}$$

where

$$\nabla^2 g = \frac{n\phi}{L_\parallel}; \tag{11}$$

the function g can be found by a further Poisson inversion and its curl taken to give a manifestly solenoidal forcing term (f_x, f_y) in the velocity evolution equation, reproducing the connection term in the vorticity update.

The code in `nektar\solvers\IncNavierStokesSolver\EquationSystems \VelocityCorrectionScheme.cpp` is modified to incorporate the above as an additional forcing term. The above modifications were inspected briefly by one of the senior developers of NEKTAR++ (David Moxey) and were judged to be correct.

2.3 Simple verification of linear evolution

The initial time-evolution of a blob in the limit of small diamagnetic drift can be calculated by a pertubative expansion, with the assumption of zero temperature diffusivity. This analysis also applies on timescales shorter than the timescales for diffusion.

Expanding about a circularly-symmetric initial solution to 4 n_0 using $n = n_0 + \epsilon n_1$, $\omega = \epsilon\omega_1$, $\phi = \epsilon\phi_1$ one has, where all quantities are the first order ones (subscript dropped) except n_0, and the prime is radial derivative,

$$\begin{aligned} \frac{\partial n}{\partial t} + \frac{n_0'}{Br}\frac{\partial \phi}{\partial \varphi} &= \frac{n_0 \phi}{L}; \\ \frac{\partial \omega}{\partial t} &= n_0' \sin\varphi + \frac{n_0\phi}{L}; \\ \nabla^2 \phi &= B^2 \omega. \end{aligned} \tag{12}$$

Now inspecting the departure from the initial condition, take all quantities to be Taylor series in time t

$$\begin{aligned} n &= n^{(1)} t + n^{(2)} t^2; \\ \omega &= \omega^{(1)} t + \omega^{(2)} t^2; \\ \phi &= \phi^{(1)} t + \phi^{(2)} t^2. \end{aligned} \tag{13}$$

One sees immediately $n^{(1)} = 0$ and the initial time evolution of $n^{(2)}$ can be determined by solving

$$\begin{aligned} \omega^{(1)} &= n_0' \sin\varphi; \\ \nabla^2 \phi^{(1)} &= B^2 n_0' \sin\varphi; \\ n^{(2)} &= \frac{1}{2}\left(\frac{n_0 \phi^{(1)}}{L} - \frac{n_0'}{Br}\frac{\partial \phi^{(1)}}{\partial \varphi} \right). \end{aligned} \tag{14}$$

It is easy to work this out for plausible circularly-symmetric blob initial states (now dropping the superscripts, and assuming $\phi \propto \sin\varphi$). The main step is evaluation of the potential from the Poisson equation

```
// sheath connection terms
// concept:
// inarray / outarray assumed to be storage for time-derivatives
// m_fields[2] is electron density n
// m_fields[3] is used to store potential phi (can't trust output as it gets advected after calculation)
// m_fields[4] is used to store solution to \nabla^2 g = n phi / L, negative curl of which is forcing term for velocity

NekDouble L = 1.0; // reconnection length

// used by HelmSolve:
StdRegions::ConstFactorMap factors;
factors[StdRegions::eFactorLambda] = 0.0;

int nPts = GetNpoints();
Array<OneD, NekDouble> dvdx(nPts, 0.0);
Array<OneD, NekDouble> dudy(nPts, 0.0);
Array<OneD, NekDouble> dummy(nPts, 0.0);
Array<OneD, NekDouble> vorticity(nPts, 0.0);
Array<OneD, NekDouble> divSheathJ(nPts, 0.0);
Array<OneD, NekDouble> velForcingX(nPts, 0.0);
Array<OneD, NekDouble> velForcingY(nPts, 0.0);

// transform elemental data if required.
for(int i=0; i<5; i++)
{
    if (m_fields[i]->GetPhysState() == false)
    {
        m_fields[i]->BwdTrans(
            m_fields[i]->GetCoeffs(),
            m_fields[i]->UpdatePhys());
    }
}

// to calculate vorticity (RHS for phi calc)
m_fields[0]->PhysDeriv(m_fields[0]->GetPhys(), dummy, dudy);
m_fields[1]->PhysDeriv(m_fields[1]->GetPhys(), dvdx, dummy);

Vmath::Svtvp(nPts, -1.0, dvdx, 1, dudy, 1, vorticity, 1);    // difference spatial derivs to form curl, sign because Hermes-3 vorticity is negative of fluid vorticity

// note: potential gets messed up later in solver so its output cannot be trusted, but it should have the right effect here
// also I think HelmSolve needs physical state fields as the forcing (assumed to be RHS) is in physical space
// phi is streamfunction (not negative of) because Hermes-3 vorticity is negative of actual vorticity
m_fields[3]->HelmSolve(vorticity, m_fields[3]->UpdateCoeffs(), factors);  // phi calc (may mess up if BCs mean there is a null space)
m_fields[3]->BwdTrans(m_fields[3]->GetCoeffs(),  //not clear why but this appears necessary after doing HelmSolve
                      m_fields[3]->UpdatePhys());

//form forcing term T phi / L
double recipL = 1.0 / L;
Vmath::Zero(nPts, dummy, 1);
Vmath::Vvtvp(nPts, m_fields[3]->GetPhys(), 1, m_fields[2]->GetPhys(), 1, dummy, 1, divSheathJ, 1);
Vmath::Svtvp(nPts, recipL, divSheathJ, 1, dummy, 1, divSheathJ, 1);  // is forcing for T=m_fields[2], is n phi/L

// now need to work out what to do to time-derivs of velocity cpts so time-deriv of vorticity gets n phi / L forcing
// add on curl source term (affects time-deriv of stream function, not that of velocity potential; sources no divergence)
m_fields[4]->HelmSolve(divSheathJ, m_fields[4]->UpdateCoeffs(), factors); // anticurl of minus (thing to add on to velocity)
m_fields[4]->BwdTrans(m_fields[4]->GetCoeffs(),  // not clear why but this appears necessary after doing HelmSolve
                      m_fields[4]->UpdatePhys());
m_fields[4]->PhysDeriv(m_fields[4]->GetPhys(), velForcingY, velForcingX);  // now velForcingX needs sign flip to form curl (sign choice is because Hermes-3 vorticity is negative)
Vmath::Neg(nPts, velForcingY, 1);

//apply forcing i.e. add on to time-derivatives
Vmath::Svtvp(nPts, 1.0, velForcingX, 1, outarray[0], 1, outarray[0], 1);
Vmath::Svtvp(nPts, 1.0, velForcingY, 1, outarray[1], 1, outarray[1], 1);
Vmath::Svtvp(nPts, 1.0, divSheathJ,  1, outarray[2], 1, outarray[2], 1);

for (int i = 0; i < 5; i++)
{
    if (m_fields[i]->GetPhysState() == true)
    {
        m_fields[i]->FwdTrans(
            m_fields[i]->GetPhys(),
            m_fields[i]->UpdateCoeffs());
    }
}
```

Figure 6: Forcing terms added into `VelocityCorrectionScheme.cpp` NEKTAR++ source file in order to give the connection terms.

$$\phi'' + \frac{1}{r}\phi' - \frac{1}{r^2}\phi = B^2 n_0'. \tag{15}$$

The initial evolution of the hot fluid blob can be inspected by running the Hermes-3 example with a small value for the diamagnetic drift term, here setting `bxcvz` 0.001, and changing `timestep` to 0.01 (this is the length of each timestep in program units) and `nout` to 20 (this is the number of timesteps to simulate) (nothing else is changed except the initial data). The vorticity is shown (called 'Vort' in the output script) for the Gaussian blob as specified above. The initial state has the expected dipolar form.

The vorticity amplitude grows linearly with time at the beginning of the calculation. In this regime, the amplitude can be calculated as

$$\omega = n_0'(r) \sin\varphi \ \mathrm{Gr}\ t \tag{16}$$

where $n_0(r)$ is the (circularly-symmetric) initial data and φ is the usual polar angle. This relation will be used to predict the initial linear evolution of `blob2d`.

A list of relevant parameters in `blob2d` is: number of timesteps `nout`$= 50$, `timestep`$= 50$, `Lrad`$= 0.05$, `Lpol`$= 0.05$, poloidal magnetic field `Bpxy`$= 0.35$, major radius `Rxy`$= 1.5$, `bxcvz`$= \frac{1}{1.5^2}$, `temperature`$= 5$.

Then the program time unit in seconds is calculated (and reported) by the program as $t_0 = \frac{m_p}{e\,\mathtt{Bpxy}} = 2.98277 \times 10^{-8}$s.

In order to access the linear regime the following modifications were made: `nout`$= 20$, `timestep`$= 0.01$, and `bxcvz`$= 0.001$. The amplitude after 10 steps will be inspected.

Now the initial data is specified as

$$n_0 = 1.0 + 0.5\ \exp(-\frac{(x-0.25)^2 + y^2}{0.05^2}) \tag{17}$$

and the gradient of this function has a maximum absolute value of $10\sqrt{2}e^{-0.5} = 8.578$. The expected vorticity amplitude in the linear regime is then, running for say 10 timesteps (and note the 0.001 is the value of `bxcvz`); also the final factor of 2 is due to an approximate formula used for $\nabla \times \frac{b}{B} \approx \frac{2}{B}$ times the curvature (thanks to Ben Dudson for this clarification)

$$\omega = 10 \times 0.01 \times t_0 \times \frac{5 \times 1.5}{0.05 \times 0.35} \times 8.578 \times 0.001 \times 2 = 2.19 \times 10^{-8}. \tag{18}$$

This amplitude is consistent with the output shown in Fig.7. In most cases it is obvious what the various factors are doing eg. the multiplier of 5 is because the diamagnetic (force) term scales with the electron temperature; the divide by 0.05 is to scale the maximum gradient (from 8.578) to the actual domain size 0.05×0.05 square metres.

Of course, all that has really been achieved is to fix a value for the product $\mathrm{Gr}\ t$, where t is the calculation duration, that applies in the linear regime. For the above computation, one has

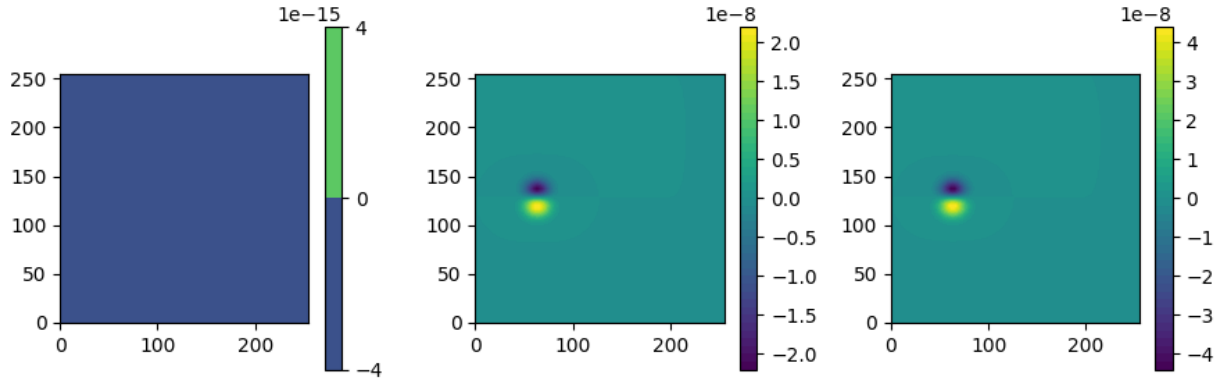


Figure 7: Plots showing the electron density after 0, 10, 20 (L-R) time steps modified as discussed in the main text. The figure for 10 steps shows amplitude agrees reasonably well with the value calculated in Eq.18. The 20-step plot confirms that the solution is within the linear growth regime.

$$\mathrm{Gr}\, t = \frac{2.19 \times 10^{-8} \times 0.05}{8.578}. \tag{19}$$

A computation matching the above initial vorticity can be performed in NEKTAR++. The calculation duration is set the same as the HERMES-3 duration in seconds ie. 2.98277×10^{-9} (100 time steps were used) and the corresponding Grashof number needs to be $\frac{2.19\times10^{-8}}{2.98277\times10^{-9}}$ divided by the maximum gradient 8.578, giving $\mathrm{Gr} = 0.85714$. In fig.8 this gives the expected amplitude. Note the vorticity is computed in PARAVIEW from the velocity components u and v output by NEKTAR++: add a Calculator for a vector quantity $u \times \mathrm{iHat} + v \times \mathrm{jHat}$ and then use the Compute Derivatives filter, selecting Vorticity for the Output Vector Type.

Now one must try to establish the value of Gr and this can be done only by matching the behaviour at longer times beyond the linear evolution regime (for now assuming that $\mathrm{Pr} = 1$). Assuming Gr takes the value above, scaled to match the force term in the original `blob2d` (ie. use $0.85714 \times \frac{1}{1.5^2\times0.001} = 380.951$) then the NEKTAR++ simulation must run for a duration of $50 \times 50 \times 2.98277 \times 10^{-8} = 7.45692 \times 10^{-5}$ in order to match the duration of HERMES-3 `blob2d`.

The problem is that the output density profile shows little change from the initial data. The suspicion is that the internal units of HERMES-3 are different to the inputs and that the code rescales the units prior to performing the time evolution and then rescaled them back afterwards (obviously this would affect the nonlinear evolution but not the linear calculation performed above). Discussions with the developers of HERMES-3 are ongoing.

2.4 Reproduction of an alternative plasma blob result from the literature

The paper [3] contains an analysis of the same sort of blob motion as in the HERMES-3 `blob2d` example. There, the Rayleigh and Prandtl numbers are made explicit hence allowing easy reproduction of the published outputs. Note the time used in that paper corresponds to time as used

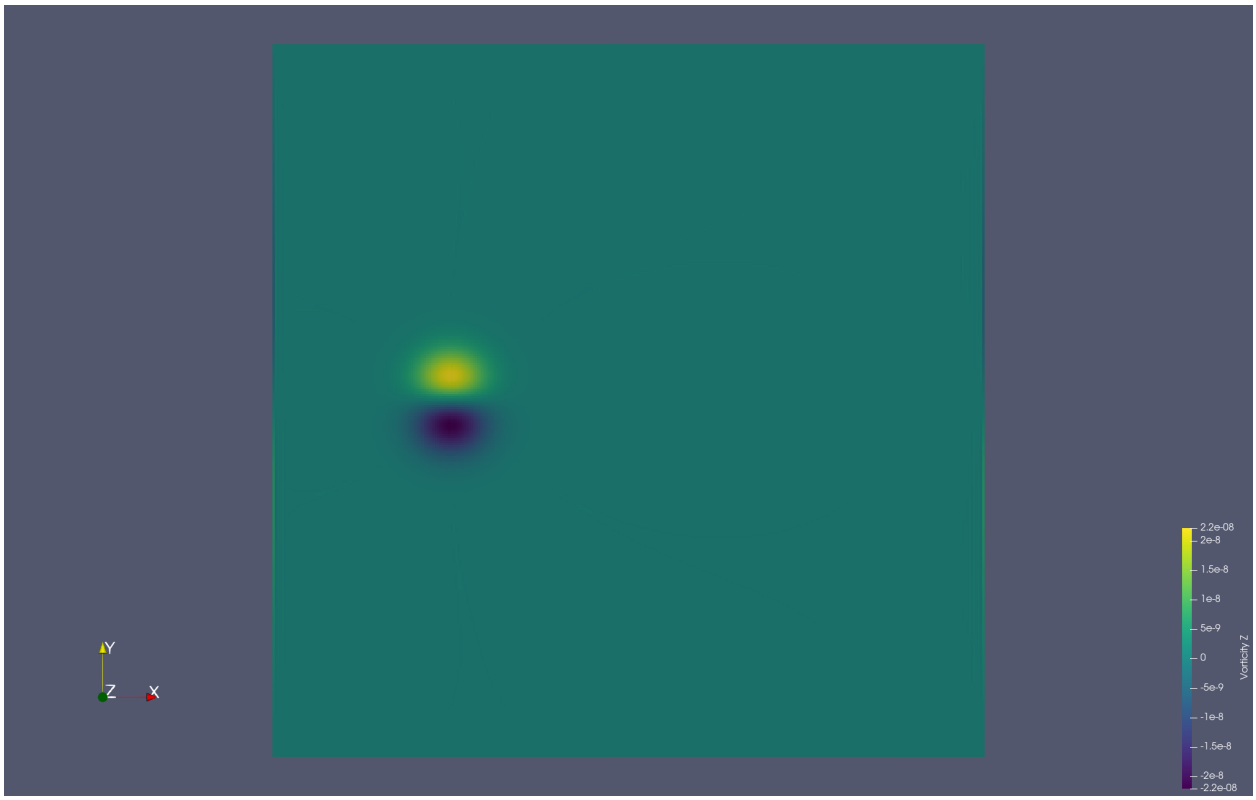


Figure 8: Plot from NEKTAR++ showing the electron density after the same time duration as in the HERMES-3 case in fig.7. The figure shows amplitude agreeing reasonably well with the value calculated in Eq.18. Note the sign reverse as indicated in the fluid mechanics-plasma analogy, Eqs.6.

in the NEKTAR++ set-up multiplied by the square root of the Rayleigh number (this can be worked out with reference to the appendix Section B).

It is straightforward to replicate the results of that paper using the incompressible Navier-Stokes solver of NEKTAR++ (Fig.9). Note that this example remains in a largely laminar flow regime and is thus not subject to the vagaries of turbulent chaos, which means the solutions match nearly exactly.

2.5 Verification of linear instability growth rate in a one-dimensional case

This subsection presents an additional fluid problem that could form the basis for a meaningful cross-validation between HERMES-3 and NEKTAR++.

Instead of the hot blob initial condition, it is possible to consider for the temperature a Gaussian density profile varying only in the x-coordinate. With the appropriate periodic boundary condition, this is then a one-dimensional system but implemented in two dimensions. In one dimension, the coupled Navier-Stokes / thermal system is

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} &= \frac{\partial^2 u}{\partial x^2} - \frac{\partial p}{\partial x} + \mathrm{Gr}\ T; \\ \frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} &= 0; \\ \frac{\partial u}{\partial x} &= 0. \end{aligned} \tag{20}$$

The final equation above is the incompressibility constraint and it restricts solutions to spatially-constant u (which gives a family of solutions related by Galileian invariance, with which in mind set $u = 0$). This means that an equilibrium of the form $\frac{\partial p}{\partial x} = \mathrm{Gr}\ T$ is a stationary solution (in one dimension, the incompressibility constraint means that heat does not rise as the fluid blocks its own movement, with the force balance being maintained by a pressure gradient countering the buoyancy). Here, the equilibrium will be taken to be a layer of hot fluid with a smooth localized temperature profile (eg. a Gaussian) added to a constant background temperature. Note that this equilibrium does not exist if there is a non-zero value of the temperature diffusivity (this would make the initial data diffuse, whatever the dimension).

In two dimensions, the equilibrium described above is still a stationary solution, but analysis shows that is not stable and eventually the instability leads to nonlinear evolution as shown in Fig.10. The associated linear instability mode and its exponential growth rate is easily calculated using NEKTAR++ or FIREDRAKE. The instability modes are seen to be localized to the region of negative temperature gradient, as expected.

The two-dimensional system of equations is

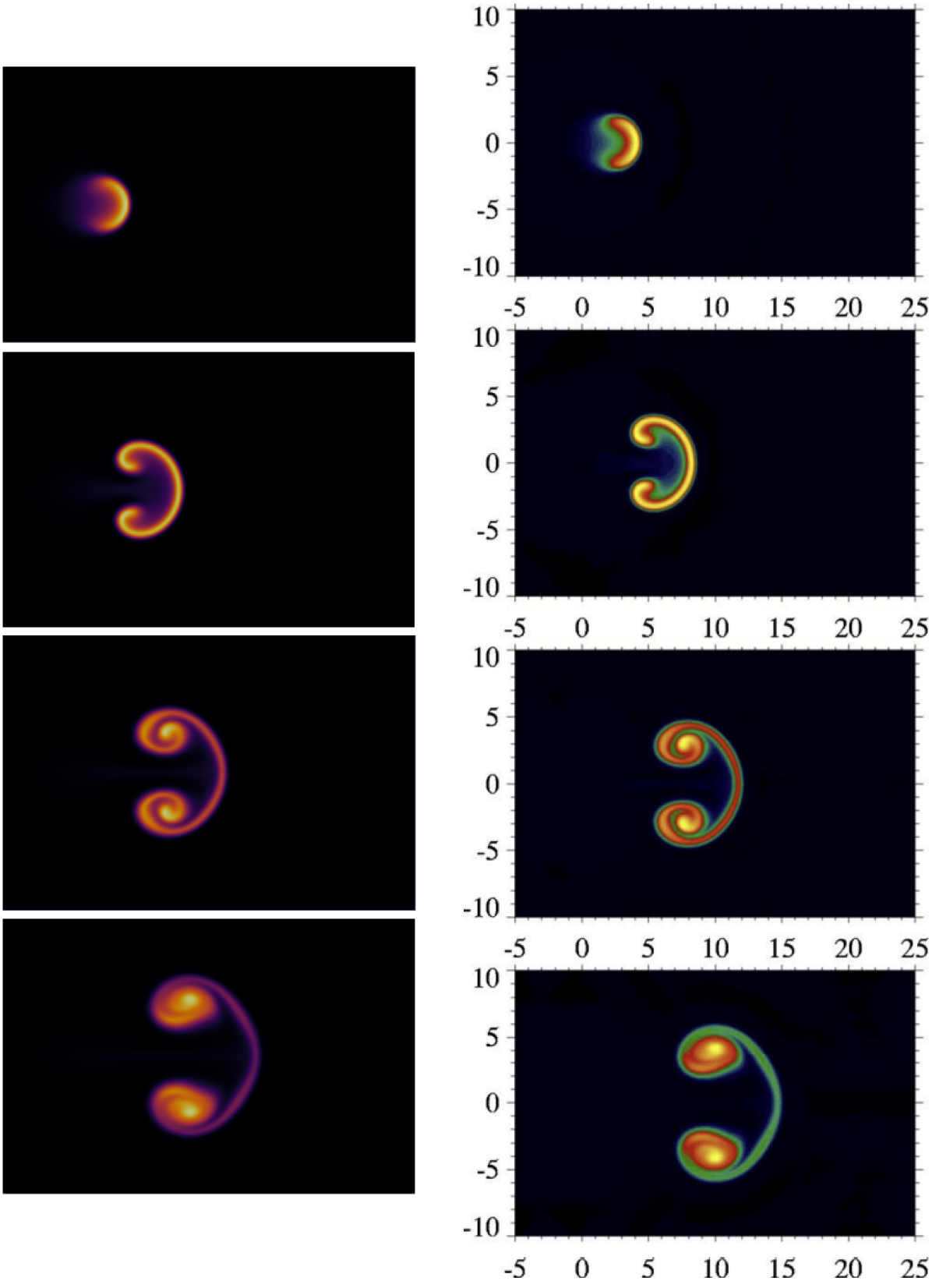


Figure 9: Plots from NEKTAR++ (left) showing time-evolution of a blob of electron density in replication of Fig.5 from [3] (right) showing very good agreement in the outputs (note the domain sizes and initial data are identical).

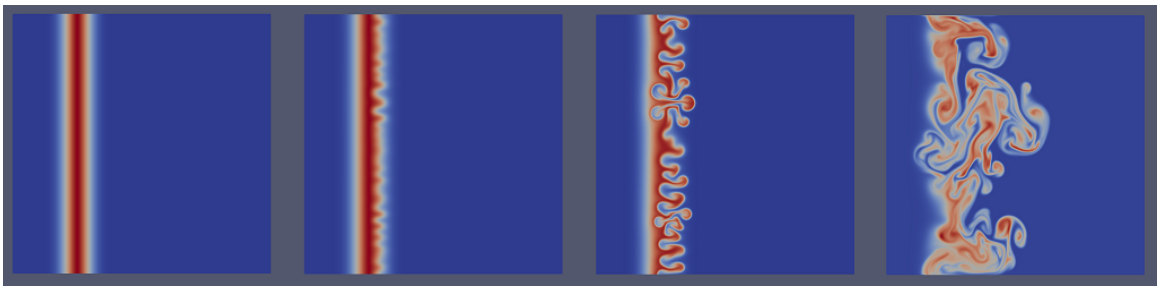


Figure 10: A sequence of frames showing to nonlinear evolution of the two-dimensional 'hot fluid layer' described in the main text.

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} &= \nabla^2 u - \frac{\partial p}{\partial x} + \mathrm{Gr}\ T; \\ \frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} &= \nabla^2 v - \frac{\partial p}{\partial y}; \\ \frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} &= 0; \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0. \end{aligned} \tag{21}$$

A linear stability analysis proceeds from the equilibrium $\frac{\partial p_0}{\partial x} = \mathrm{Gr}\ T_0$ (all other quantities being zero), giving the equations for the linear perturbation

$$\begin{aligned} \lambda u &= \nabla^2 u - \frac{\partial p}{\partial x} + \mathrm{Gr}\ T; \\ \lambda v &= \nabla^2 v - \frac{\partial p}{\partial y}; \\ \lambda T + u\frac{\partial T_0}{\partial x} &= 0; \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0. \end{aligned} \tag{22}$$

By assuming separability and assuming wavenumber k in the periodic y-coordinate, one can obtain a higher-order equation for the x-component of velocity u as

$$k^2\left(\lambda + k^2 + \frac{\mathrm{Gr}}{\lambda}\frac{dT_0}{dx}\right)u = -\frac{d^4u}{dx^4} + \lambda\frac{d^2u}{dx^2} \tag{23}$$

(note that this is almost reducible to a Schrödinger equation: differentiate twice $Eu = -\frac{d^2u}{dx^2} + Vu$ to $\frac{d^4u}{dx^4} + E\frac{d^2u}{dx^2} = \frac{d^2V}{dx^2}u + 2\frac{dV}{dx}\frac{du}{dx} + V\frac{d^2u}{dx^2}$ and observe only the single derivative term is discrepant). Note also that if $k = 0$ then the coupling to the background temperature gradient vanishes - this corresponds to the (stable) 1-D case where there can be no y dependence.

One can look for scaling laws by taking appropriate limits; as an example, large λ and constant $\alpha = \frac{\mathrm{Gr}}{\lambda^2}$ gives the equation

$$\frac{d^2u}{dx^2} = k^2 \left(1 + \alpha \frac{dT_0}{dx}\right) u \tag{24}$$

which is suggestive of the scaling law $\lambda \sim \mathrm{Gr}^{\frac{1}{2}}$ (and which clearly has purely real eigenvalues due to self-adjointness), and small λ and constant $\beta = \frac{\mathrm{Gr}}{\lambda}$ gives

$$-\frac{d^4u}{dx^4} = k^2 \beta \frac{dT_0}{dx} u \tag{25}$$

which is suggestive of the scaling law $\lambda \sim \mathrm{Gr}$. The scaling laws are observed in Fig.11. Note these scalings are useful in determining estimates for the eigenvalues for inputs as initial guesses for numerical eigenvalue computations, which can be dramatically faster for good initial data.

The two-dimensional case was simulated in the incompressible Navier-Stokes solver of NEKTAR++. The initial temperature condition was exchanged for a Pöschl-Teller-like (sech^2) profile so that the initial pressure profile can be provided analytically to NEKTAR++ which currently supports expressions provided by the cmath library (for example, $\mathrm{erf(x)}$ is not available); the temperature profile was specifically $T = \frac{1}{2}\mathrm{sech}^2\frac{x}{a}$ with $a = 0.05$ and no additional perturbation was applied. For a cross-validation, it was found that simulations performed for a selection of values exhibited linear growth rates largely consistent with those output by FIREDRAKE. Note the NEKTAR++ growth was tracked using the growth of the kinetic energy (because this involves a quadratic function of the mode, the energy growth rate is twice the linear coefficient); it was necessary to modify the `FilterModalEnergy` filter of NEKTAR++ for this example as the supplied version counts any advected scalars (here temperature) as additional components of velocity and so misreports the fluid kinetic energy. A word on temperature diffusivity in the NEKTAR++ incompressible Navier-Stokes solver is also indicated in that it is necessary to specify the diffusivity for the T field (which is formally an additional advected scalar in the solver) explicitly in the session file, or a (unit) default value will be applied which causes diffusion in the initial condition. The appropriate lines are

```
<FUNCTION NAME="DiffusionCoefficient">
<E VAR="T" VALUE="1.0e-8" />
</FUNCTION>
```

It is not possible to specify precisely zero diffusivity here as this results in a numerical error (as noted earlier, the time-evolution scheme involves division by the diffusivity). The NEKTAR++ simulations described in this section were therefore performed by taking a small value (10^{-16}) for the thermal diffusivity.

This problem is related to stability analyses of Rayleigh-Bénard convection first performed by Lord Rayleigh. Note that the example here is not stable for negative values of the Grashof number (unlike Rayleigh's example) as it has a positive as well as a negative temperature gradient region; the stability characteristics are actually invariant under a change in the sign of Gr. Along similar lines, a layer of this type is always unstable, whereas the Rayleigh case is stable up to a certain critical value of Rayleigh number Ra (which is independent of the Prandtl number) beyond which the system undergoes a Hopf bifurcation. Rayleigh's problem is also different in having a finite

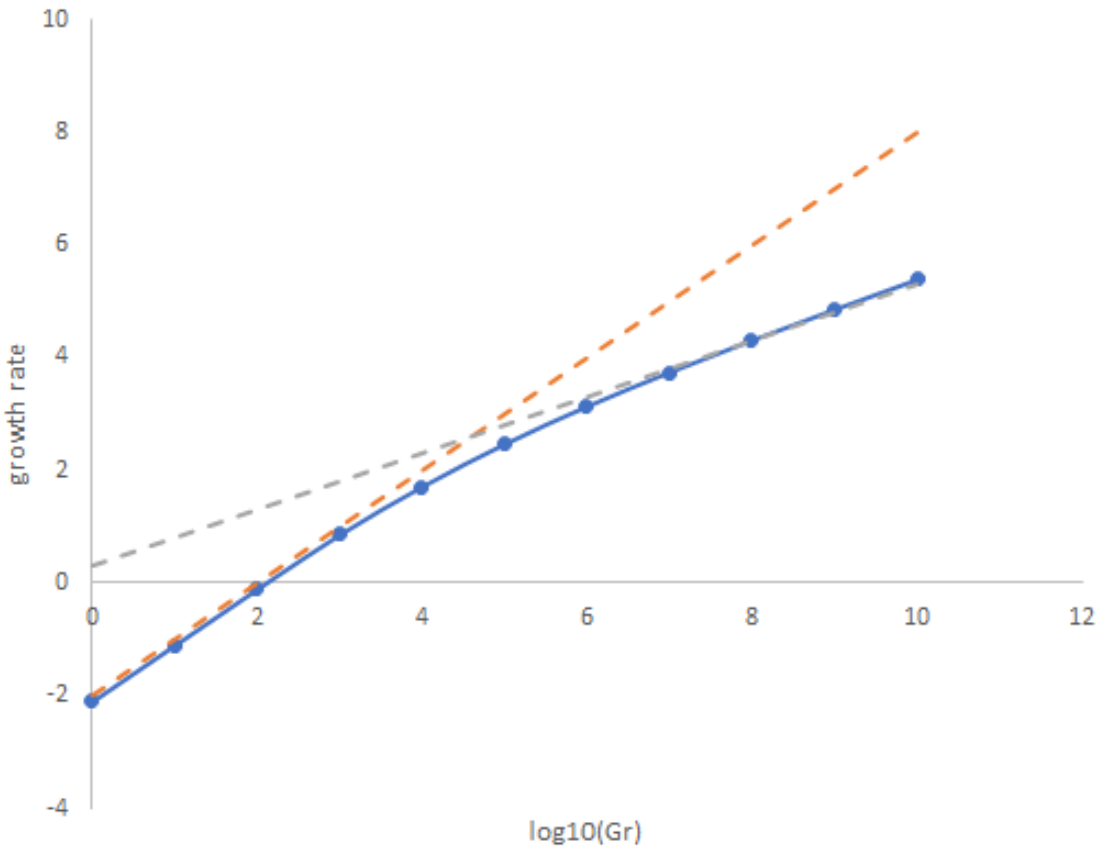


Figure 11: The dependence of the linear instability growth rate for the hot layer on the Grashof number Gr. The orange dashed line is a linear fit, which applies at small Gr; the grey dashed line has gradient $\frac{1}{2}$ and corresponds to the square-root dependence observed at large Gr.

Prandtl number, which cannot be introduced here as it would cause the initial data to diffuse (the problem discussed here exists in the $Pr \to \infty$ limit).

It is straightforward to run the FIREDRAKE eigen-analysis repeatedly and thus obtain the largest growth rate as a function of the Grashof number. This can also be done using the time-domain incompressible Navier-Stokes solver of NEKTAR++ though this is a little less convenient due to the need to adjust the calculation duration and timestep size for each simulation (the linear regime must be reached from an initial quiescent state and the timestep size must be such that the simulation is stable). A comparison of numerical eigenvalues obtained using both frameworks is exhibited in Table1. Such analysis reveals the dependence of the largest eigenvalue on Gr (Fig.11) and that this scales according to the power laws described earlier.

A more detailed inspection of this curve reveals discontinuities in the derivatives when the most unstable mode changes values of the transverse wavenumber k (Figs.12, 13); the system exhibits a response surface of discontinuous gradient.

The same problem can be studied in three dimensions and in that case there is a richer spectrum of transverse harmonics determined by the container geometry; see Fig.14. The 3-D case is more

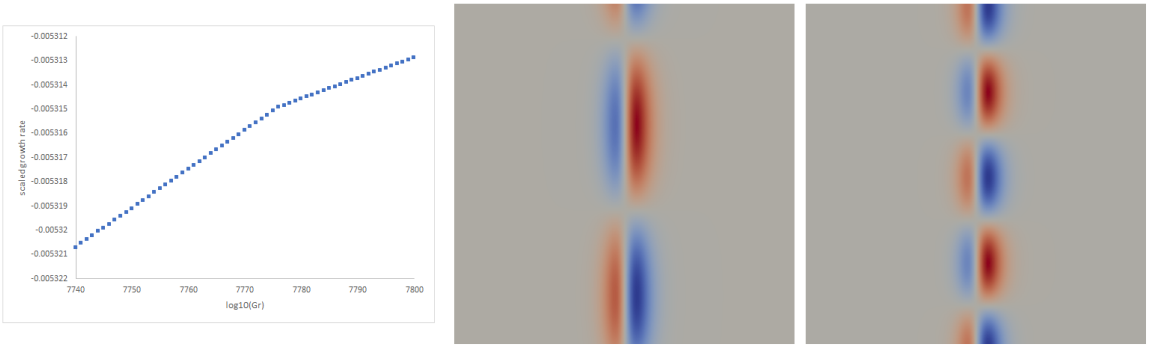


Figure 12: The dependence of the linear instability growth rate for the hot layer on the Grashof number Gr near the transition between the transverse $k = 1$ and $k = 2$ modes (the two colour maps correspond to the temperature component of the mode at $\mathrm{Gr} = 7000$ (leftmost colourmap) and $\mathrm{Gr} = 8000$ (rightmost)) .

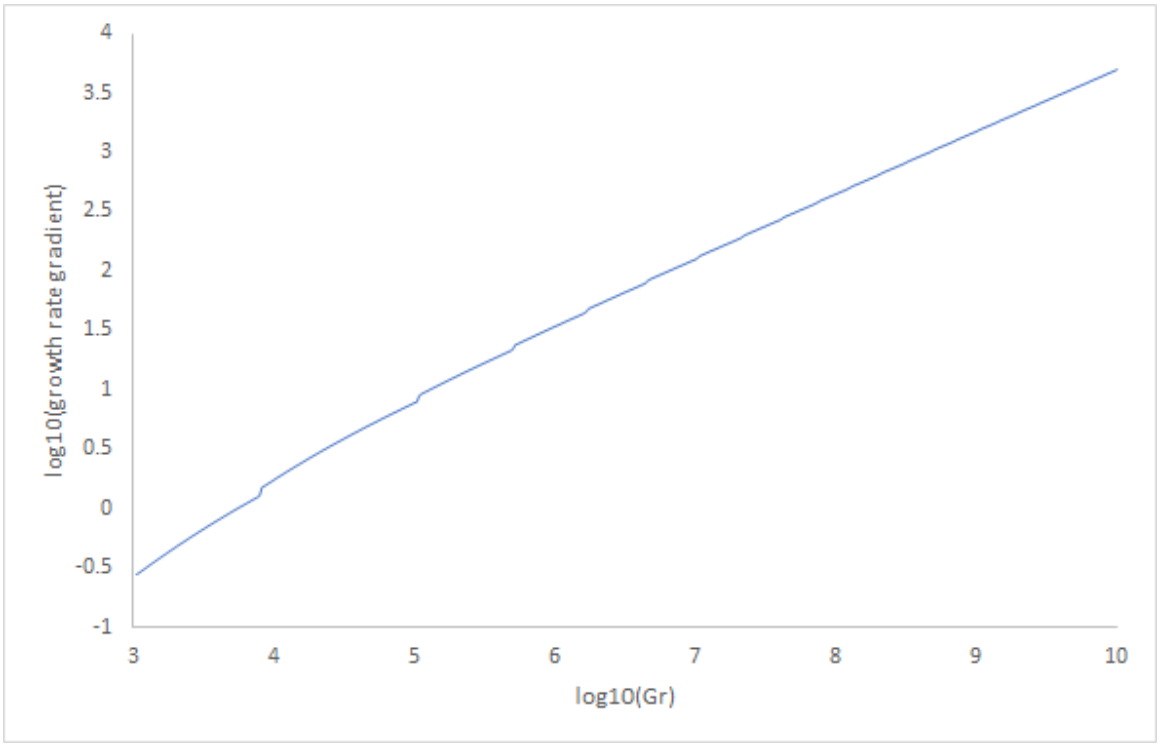


Figure 13: The dependence of the linear instability growth rate for the hot layer on the Grashof number Gr: the difference between subsequent values has been taken in order to show the derivative discontinuities as jumps; these occur at a change in the transverse wavenumber k of the most unstable mode.

$\log_{10}\mathrm{Gr}$	λ (FIREDRAKE order $3,2,3$)	λ (NEKTAR++)
0	0.007810	0.035
1	0.07803	0.0827
2	0.7739	0.7993
3	7.198	7.3944
4	51.38	50.478
5	293.00	292.22
6	1368.70	1358.3
7	5511.75	5449.2
8	20418.12	20283
9	71489.00	70506
10	241978.31	238642

Table 1: Table of linear growth rates for the hot-layer problem. The NEKTAR++ growth rates are seen to be slightly less than the FIREDRAKE ones; one reason is the need to include a small amount of temperature diffusivity in the NEKTAR++ simulations; also the way the boundary conditions are done differs somewhat between the two codes. The NEKTAR++ value for $\mathrm{Gr} = 1$ is clearly discrepant.

challenging with only, say, a 10^3 element mesh with element orders up to three being feasible on a 32Gb machine.

It is hoped that examples of this type can provide a cross-validation of simulations performed using HERMES-3. The results obtained lie within the linear regime of the simulations and are thus not subject to the chaotic aspects of nonlinear evolution. It is also the case that the early time-evolution studied here is independent of the details of viscosity or temperature diffusivity (both of which operate on longer timescales than treated here).

Actually performing the study for HERMES-3 has been delayed due to the aforementioned problems in clarifying the units used in HERMES-3.

2.6 Space- and time-dependent source terms in NEKTAR++

In view of the need to implement source terms in NEKTAR++ (eg. for coupling to a particle-based or other kinetic code), a brief test was made of the existing capabilities of the code to handle forcing terms.

The buoyancy force in the incompressible Navier-Stokes solver is enabled in the session file using the `BodyForce` capability; this corresponds to the top line in the sequel. It is also possible to include space- and time-dependent source terms into these directives, but the `EVARS` section must be left out as this forces the code to look up the equation variables for the forcing (as done for the buoyancy force) and to ignore the space / time variables (here `x, y, t`).

```
<FUNCTION NAME="BodyForce">
<E VAR="u" VALUE="Gr*T" EVARS="u v T p" />
<E VAR="v" VALUE="0" EVARS="u v T p" />
```

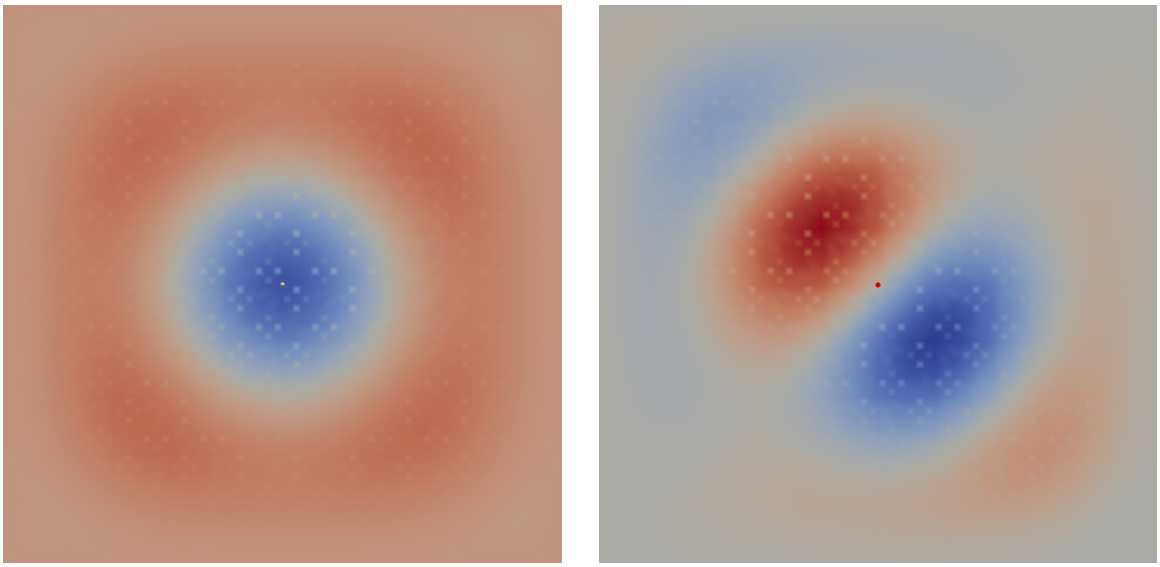


Figure 14: Transverse temperature profiles of instability modes for $\mathrm{Gr} = 7 \times 10^3$ (left) and $\mathrm{Gr} = 2 \times 10^4$ (right). The simulations used orders $3, 2, 2$ for velocity, pressure, and temperature. The plot artifacts are due to the rather low resolution. Note that these simulations used Dirichlet conditions at the transverse boundaries, due to the lack of a partially-periodic cube mesh in FIREDRAKE. (Plots obtained using PARAVIEW slice view at $x = 0.55$ to coincide with steepest part of unstable gradient.)

```
<E VAR="T" VALUE="(1.0e4)*exp(-((x+0.25)^2+(y+0.25-5000.0*t)^2)/(0.05^2))" /> <!--
space- and time-dependent forcing term -->
</FUNCTION>
```

The output is shown in Fig.15.

2.7 Concluding remarks

The above sections described work aiming to replicate some of the outputs of the established plasma fluid code HERMES-3 using NEKTAR++. Initial outputs show some similarity in the be-

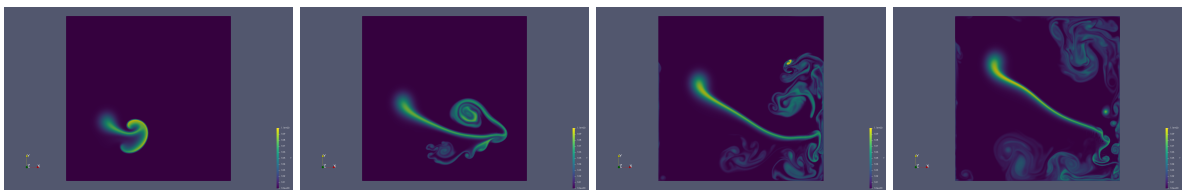


Figure 15: Time-evolution of a simulation with a heat source that translates upward during the simulation. Note that numerical errors are evident: the third frame shows a blob of excessively large and small density.

haviour of plasma blob-type simulations but the analysis is complicated by difficulty in clarifying the units used in the HERMES-3 code, and also the fact that the viscosity and temperature diffusivity in HERMES-3 are implemented numerically and exhibit grid-dependence.

This work is in progress and it is currently intended that any doubt as to the units be removed by running the code within a debugger in order to expose exactly what system is being solved.

3 Analysis of vertical convection in FIREDRAKE

It is straightforward to simulate at least time-independent convection problems using the FIREDRAKE PDE solver framework. This restricts the study here to the laminar regime, though it is of course possible to implement a time-stepper. Here a vertical natural convection problem is studied; a fluid-containing cavity transfers heat from a hot side to a cold one. The parameters of the problem are the dimensionless temperature difference (Rayleigh number Ra) and the usual Prandtl number Pr of the fluid.

This work provides an additional cross-validation against NEKTAR++ simulations of the same problem as well as validation against recent results in the literature; the model used here follows the set-up in [8] with a unit square cavity and a value of $\mathrm{Pr} = 10$ (the latter is not far from the value for water, approx. 7).

3.1 Calculation of Nusselt number

The Nusselt number is easily calculated using a few lines of Unified Form Language (UFL, used in the FIREDRAKE interface). In the sequel, normals `normL` and `normR` are constant members of the vector space V, the `assemble` command calculates the components of the argument (scalar here), and `ds(1)` and `ds(2)` refer to the composites labelled by 1 and 2, which are respectively the left- and right-hand boundaries and are domains of integration for computing the Nusselt number, which is then `0.5*(fluxL-fluxR)`.

```
normL = Function(V)
normL = Constant((-1.0,0.0))
fluxL = assemble(inner(normL, grad(T))*ds(1)) normR = Function(V)
normR = Constant((1.0,0.0))
fluxR = assemble(inner(normR, grad(T))*ds(2))
```

This high-level functionality can be compared with adding the same functionality to a NEKTAR++ simulation, in which a Filter needs to be added to the source code and the relevant parts of the code recompiled; the filter must then be activated by adding a command in the session file. As a reference, the `AeroForces` filter included with NEKTAR++ codebase calculates the aerodynamic forces on a surface or body and can be easily modified to compute heat fluxes (the main step is to change the relevant field to the temperature and to take the derivative); the file `FilterAeroForces.cpp` is of order 1000 code lines in size.

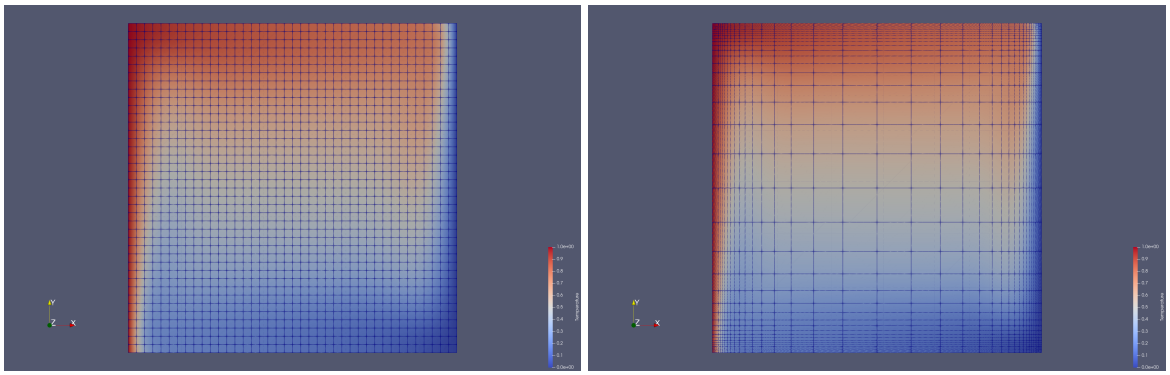


Figure 16: Steady-state temperature profiles from the case discussed in the text for $\mathrm{Ra} = 7 \times 10^5$ and element orders $2, 1, 1$ on a uniform grid (left) and for $\mathrm{Ra} = 2 \times 10^6$ and element orders $5, 4, 4$ on the boundary-refined mesh. The meshes are visualized for illustrative purposes.

3.2 Simple continuation method

An initial study involves setting up the problem with a direct solver (as described in [9]) and proceeding to solve the equations with zero values in the initial state. Using order-two elements for the velocity components and one order less for the pressure and the temperature (this, called here order $2, 1, 1$ is the usual Taylor-Hood element choice) and a uniform 40×40-element grid, it is found that the method works well up to $\mathrm{Ra} = 7 \times 10^5$ but that it fails for $\mathrm{Ra} = 8 \times 10^5$ and above.

It is found that increasing element order gives only very small improvement in the maximum value of Ra for which the simulation can be performed successfully: increasing order to $4, 3, 3$ gives convergence for $\mathrm{Ra} = 8 \times 10^5$ but not $\mathrm{Ra} = 9 \times 10^5$.

Increasing element order in conjunction with the boundary-refined mesh it is possible to do a little better with $\mathrm{Ra} = 2 \times 10^6$ converging (but $\mathrm{Ra} = 3 \times 10^3$ not) for element orders $5, 4, 4$. This improvement is clearly not sufficient to reproduce results from [8] which extend up to $\mathrm{Ra} = 10^{14}$ using a finite-difference code (a more relevant target here is $\mathrm{Ra} \approx 5 \times 10^{10}$, beyond which it is found [8] that the flow ceases to be laminar and stationary).

The boundary-refined mesh was used with a simple continuation method to reach higher values of Ra. Here, a sequence of simulations is performed for gradually-increasing values of Ra, using the result of the previous calculation to initialize the solver for the next one. It was found that an incremental factor of $\sqrt{10}$ starting from $\mathrm{Ra} = 10^5$ gives a method that works up to and beyond the limit of the laminar regime. (It was, however, noted that very high values of Ra (beyond say 10^{12} sometimes resulted in out-of-memory errors on a 32Gb machine.) This gives an efficient method for obtaining the physically-interesting relation $\mathrm{Nu}(\mathrm{Ra})$ (the formulation as an elliptic problem removes the need to simulate until any transients have decayed). The relation shows the expected power-law scaling $\mathrm{Nu} \sim \mathrm{Ra}^{\frac{1}{4}}$ characteristic of the laminar flow regime, Fig.17. Good agreement with the values obtained by [8] is seen up to where the laminar regime ceases to be the true time-domain solution.

$\log_{10}\mathrm{Ra}$	Nu (Wang *et al*, [8])	Nu (orders $4,3,3$)	$\mathrm{Nu}(5,4,4)$	$\mathrm{Nu}(6,5,5)$
7	17.45	17.33	17.33	17.33
8	32.17	31.78	31.79	31.79
9	57.83	57.44	57.53	57.56
10	103.57	103.57	103.04	103.33

Table 2: Table of steady-state Nusselt number values. The values from [8] are the highest grid resolution ones quoted.

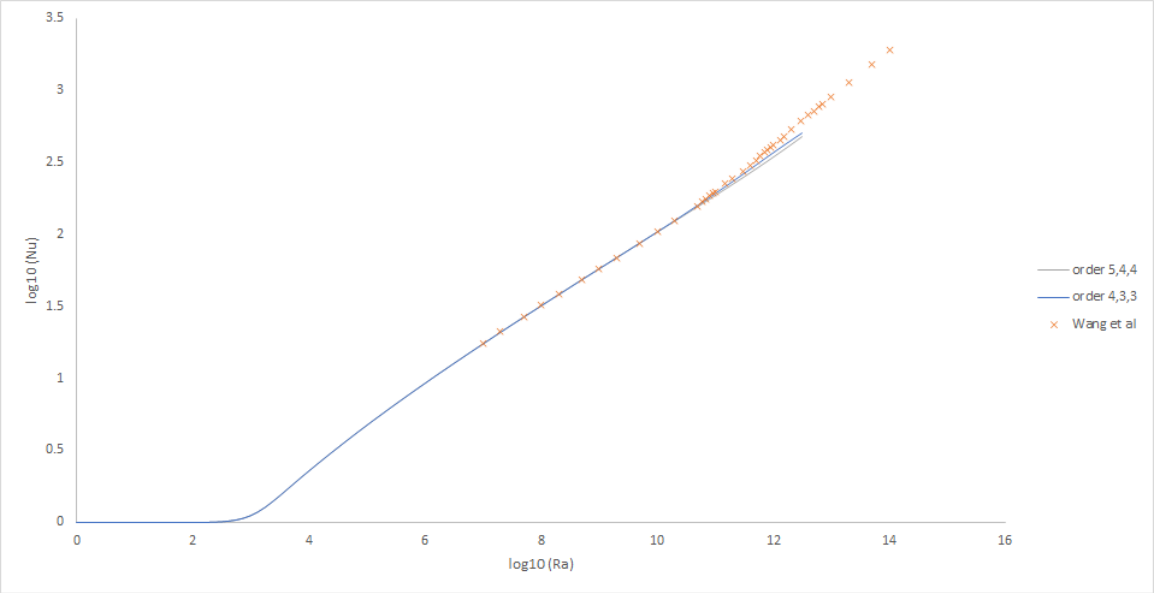


Figure 17: Graph of heat transfer rate represented by Nusselt number Nu as function of Rayleigh number Ra. Two datasets from FIREDRAKE are shown, using elements of order $5,4,4$ and $4,3,3$ respectively - there is a small amount of discrepancy for the largest Rayleigh numbers simulated. These outputs can be compared with the results from Wang et al, showing agreement up until the emergence of the time-dependent state for $\mathrm{Ra} \geq 5 \times 10^{10}$ where the scaling changes from the classic $\frac{1}{4}$ laminar power law to the $\frac{1}{3}$ power law associated to Malkus scaling [10].

3.3 3-D version

It is possible to perform the same calculation in 3-D in order to access the laminar flow regime of a real tank (with three-dimensional no-slip boundary conditions). These 3-D flows are relevant to the *Smallab* experiments being performed at the University of Leeds. Note that steady flows in a 3-D geometry with periodicity in the extra dimension will be identical to those in two dimensions, though the linear stability properties would be expected to differ.

4 Task work by grantees

Accepted reports are held on the Documents repository of the ExCALIBUR-NEPTUNE organisation on GitHub, to which belong other repos containing software developed under project NEPTUNE. Note that some of these repos may be access-controlled, please email `neptune@ukaea.uk` if difficulties are encountered in seeing the material.

4.1 Reports received under Grant T/NA078/21, PO (King's College London)

- 2053622-TN-03-3 - D1.2: Augmentation of the NekMesh generator to provide quad-based meshes for 2D configurations [11]
- 2053622-TN-04-2 - D2.4: Evaluation of preconditioners of matrix-free operations in Nektar++ for anisotropic heat transport [12]

4.1.1 2053622–TN–03

Report 2053622–TN–03 concerns NEKMESH, NEKTAR++'s built-in meshing utility. It describes some of the current capabilities of the tool and an extension that will allow it to generate 2-D meshes composed of purely quadrilateral elements. In previous version of NEKMESH, quadrilaterals could only be added as part of the boundary layer meshing functionality. "Quad" meshes have some important computational advantages over triangular meshes; in particular, they tend to use fewer elements overall and provide superior performance when using matrix-free operators. This latter factor is expected to be of critical importance when applying high-order methods on exascale platforms, as a matrix-free approach can reduce memory bandwidth and greatly increase computational intensity. The same arguments also apply to hexahedral meshes in 3-D, though the new developments discussed in this report are for 2-D applications only.

In section 3, the authors describe NEKMESH's existing quad/hexahedra construction capability, which is used in *isoparametric boundary-layer generation*. This technique extends a given boundary (mesh constraining curve) in the normal direction, thus creating a layer of quadrilateral elements either inside or outside said boundary. The authors concede that the existing implementation does not consider interactions between nearby curves when treated in this manner (which could create problems when adding boundary elements to nested flux surfaces, or x-point geometries). They also explain that a further difficulty exists with adding boundary layers to both

sides of an embedded constraint curve (as opposed to a boundary curve), though a fix for this is currently in-development. Another existing capability is to refine the boundary layer by splitting the quads along the direction transverse to the surface normal; the splitting can be graduated so that the element density in the normal direction increases geometrically toward the boundary (one current limitation here is that this also can only be done on one side of a constraint curve). These methods are compatible with the curved meshes, on which the splitting scheme relieves difficulties inherent in more general approaches to anisotropic boundary refinement. Subsection 3.3 gives an instructive illustration of the refined boundary layer (on a 3-D mesh).

Section 4 describes a new method for generating fully-quadrilateral meshes. This code currently exists on a work-in-progress branch of the NEKTAR++ repository, and is the subject of a recent publication. The core of the method is the use of a systematic approach to detect where non-valence-four vertices are located. The authors plan to validate this approach for tokamak-relevant geometries and to integrate it into the NEKMESH workflow, though they expect it will only be applicable to 2-D use cases.

Finally, the authors conclude that structural changes to the NEKMESH code will be necessary in order to proceed beyond the current version, which was designed with a focus on aeronautical applications eg. isolated aerofoils in empty space. They suggest a number of technical changes required to fully integrate the new method in NEKMESH and allow others in project NEPTUNE to make use of the functionality.

4.1.2 2053622–TN–04

Report 2053622–TN–04 provides a summary of the preconditioner types currently available in NEKTAR++ and describes work to develop a new preconditioning method. It goes on to present a comparison of the different options based on their performance when applied to a steady-state, two-dimensional, anisotropic diffusion problem.

A formulation of the benchmarking problem is presented in Section 2. The authors begin by explaining the form of the diffusivity tensor in a magnetised plasma, then derive the anisotropic thermal conductivity tensor that features in the problem definition. Section 2.1 describes how the problem is discretised in variational form and expressed as a matrix system.

Implementation details of NEKTAR++'s existing preconditioners are discussed in Section 3, along with a new method called 'Geometrically Informed Algebraic Multigrid' (GIAMG), which uses the *Saena* library. As background, the authors explain that the algebraic multigrid (AMG) method is well suited to unstructured meshes, but tends to suffer from a loss of sparsity which can affect performance and result in poor scaling. The new method addresses this weakness by using a combination of 'p coarsening' (polynomial order reduction) and 'h coarsening' (coarsening of the mesh itself).

Results of the comparison study are presented in Section 4. Performance (iteration count or run time) is presented as a function of both anisotropic ratio and transport angle for the full matrix system and for the 'statically condensed' version. In both cases, the best results are achieved by the 'Diagonal' preconditioner. The new GIAMG method typically takes many fewer iterations to arrive at a solution, but tends to have longer solve times overall. The authors point out, however,

that GIAMG is expected to scale better than the linear space (XX^T) solver for large problems ($> 10^4$ cores) and that the benchmark problem does not approach this scale. They also note that they expect the efficiency of the new method to improve with further development.

4.2 Reports received under Grant T/NA085/21, PO 2047357 (University of Oxford)

- 2047357-TN-10 - Numerical study of 1+1D drift kinetic models for parallel dynamics in the plasma edge [13]

- 2047357-TN-13-2 - Numerical study of a reduced model coupling 2D+2V drift-kinetic ions and 2D+3V kinetic neutrals in a helical magnetic field with wall boundaries [14]

- 2047357-TN-14-2 - Overview of the numerical issues and findings associated with the 1D and 2D drift kinetic models [15]

4.2.1 2047357–TN–10

The report presents the various $1d1v$ versions of the moment kinetic approach, separating out different numbers of fluid moments in the sense that first only density is treated separately from the rest of the particle distribution for the species, then density and velocity, and finally all the five special moments of density, velocity (counts by itself as 3 moments) and energy are treated separately, effectively giving a series of coupled fluid and kinetic models. For each of the three cases, the corresponding boundary conditions are specified. Coordinates are made dimensionless (using the standard project notation rather than that of the report) based on a choice of L_s, U_s and employing N_{ref} to normalise number density.

The report then presents Julia-derived numerical results for the various cases, using SSPRK (Strong Stability Preserving Runge-Kutta) schemes for the time advance and either third order upwind finite difference schemes or Chebychev polynomials to represent the spatial dependence. The latter use the Gauss-Chebyshev-Lobatto grid that has nodes at element edges. Conservation of integral quantities is enforced by uniform scaling of the moments and distribution functions at each timestep. Realistic boundary conditions at walls, as well as the usual (largely irrelevant outside the plasma separatrix) periodic conditions, are implemented.

Simulations are presented of a wall-bounded model and a periodic model. The former is designed to compare with an analytic model of the plasma sheath, and the latter with a Landau-damped sound wave, ie. a sound wave which kinetic effects cause to be damped. These simulations exhibit spectrally accurate convergence, ie. that errors diminish exponentially with number of mesh-points, demonstrating the correctness of the implementation of the model and the spatial and temporal discretizations. There is discussion as to the implementation of plasma-neutral collisions, in which there is argued the need for a separate grid in velocity space for the neutrals.

4.2.2 2047357–TN–13–2

This report extends the work of ref [16] to two dimensions in space and two or three dimensions in velocity space, ie. $2d2v$ or $2d3v$. Only the moment corresponding to fluid density is treated separately. In respect of the two space dimensions (r, z), r corresponds to distance from the plasma centre and z to distance along the axis of a circular cylinder. The magnetic field, has a fixed poloidal component (ie. directed in the ζ direction - for which θ is the preferred symbol) as well as having the expected component in z, which is also taken to be constant throughout the model. Poloidal, ie. ζ, variation of quantities is not allowed. The same normalisation is used for the equations to make them dimensionless. Ion-neutral collisions are however now included.

Computationally, the same language (Julia) and the same numerical approach is used as before, except that the spatial distributions are now represented exclusively as Chebychev polynomials, ie. no finite differences. An important point is the use of the electron Boltzmann response relation $n_e \propto \exp(q_e\Phi/T_e)$ (in standard project notation) to calculate the electric potential Φ from n_e and hence the electrostatic field that acts on the plasma. The neutrals are represented on a 3-D grid which, since the neutrals are unaffected by electromagnetic forces, ignores the presence of magnetic field. (There is an interesting discussion of the issues involved in having different 'best' representations for charged and neutral particles respectively.) Implementation details for the MMS (see next paragraph) are presented.

Analytically, a solution for the plasma sheath is presented, but most of the solutions derived for testing the numerics are derived by use of the Method of Manufactured Solutions (MMS). MMS may be illustrated using a simple ODE, viz.

$$\ddot{y} + \Omega^2 y = 0 \tag{26}$$

MMS entails intelligent selection of $y(t)$, then forming a source term $S(t)$, to appear on the right-hand side of the equation, then solving

$$\ddot{y} + \Omega^2 y = S(t) \tag{27}$$

For example if is chosen $y(t) = \cos(\omega t)$ so that

$$S(t) = (\Omega^2 - \omega^2)\cos(\omega t) \tag{28}$$

then there should be zero residual if the modified equation is solved with this form of additional source term. There are potential difficulties, in that the numerical work may be of little value if $y(t)$ is inappropriately chosen, thus a sinusoidal forcing may not tell much about solutions of

$$\ddot{y} - \Omega^2 y = 0 \tag{29}$$

which has exponentially growing solutions. Moreover, resonant effects in Equation (26) could cause difficulties if ω happened to have been chosen close to the natural frequency Ω.

As in the $1d1v$ work [16], many simulations, including all those with periodic boundary conditions, demonstrate spectrally accurate convergence. However, when an electric field is present, other simulations demonstrate not only poor convergence, but (in private communications) indications of divergence. There follows an honest report of numerical difficulties encountered at the frontiers of computational plasma physics. Possible causes of the difficulties are discussed at more length in the follow-up report [15].

4.2.3 2047357–TN–14–2

Work in the report [14] showed that problems arose in 2-D kinetic modelling tested using MMS, when a wall boundary condition was applied in the z coordinate. The report [15] which follows, discusses a number of possible causes of these problems (but note that references [7] and [8] need correcting to cite reports 2047357-TN-10 and 2047357-TN-13 respectively). Very recent work has served to identify the prime cause and its elimination, so the report below is largely superseded. The manufactured solutions might have been expected to be of wider value, but subsequent experience has indicated that they are the primary cause of the numerical difficulties encountered.

It is worth however noting that sheath conditions give rise to a boundary condition on velocity depending on the value of the electric field, and in a kinetic model, velocity is itself a coordinate over which the unknown distribution function f is defined. The upshot is that a condition for f to vanish is applied between nodes, so that f is not smooth within the computational domain and spectral convergence was impossible without employing a smoothing algorithm. In addition, the MMS numerical tests (private communication) indicated an even worse problem in that there was not only poor convergence, but indications of divergence. It was acknowledged there and below that dealing with this would be challenging, because there was then no certain identification of the cause of the problem(s). The report below [15] also notes that, regardless of the original research plan, resolving the numerical difficulties has to be the priority objective, a decision vindicated by subsequent developments.

4.3 Reports received under Grant T/AW086/22, PO 2070839 (University of Oxford)

- 2070839-TN-01 - Update on state of the art in edge modelling [17]
- 2070839-TN-02 - Implementing distributed-memory in the 2D-3V drift-kinetic edge code [18]

4.3.1 2070389–TN–01

This report updates the earlier document [19], specifically the review of the state-of-the-art in edge modelling is brought up-to-date. The updated report focusses on developments which the authors believe to represent an advance over previous edge modelling work, especially if they involve kinetic treatments. These leads to the discussion of approximately a dozen new citations, mostly dating from the last years 2021-2022. The advances turn out all to be concerned with the addition of further physical effects, mostly to the codes directed at modelling kinetic effects.

There is a description at the end of the still severe limitations of most kinetic modelling in treating effects at the transition from closed to open fieldlines at the plasma edge, ie. across the plasma boundary defined by separatrix, and particularly near the X-point where the poloidal component of magnetic field vanishes. The authors expect a special treatment to be needed close to the latter place. The hope, in the context of the broader project, is that unstructured finite element meshes will not only enable solution near the X-point, but might be suitably adapted (ie. in conformity with

the magnetic field direction) so that co-ordinates which explicitly follow fieldlines can be avoided altogether.

4.3.2 2070839-TN–02

The report describes the further parallelisation of the $2d3v$ Julia software under development at Oxford, in order to speed turnaround of tests of the software and thereby lead to faster identification of the problem issues. The additional parallelisation sees implementation of distributed-memory MPI alongside the shared-memory MPI that was already in use; both MPI features use the Julia MPI library. The resulting source code is made available on github.

The distribution of computational nodes applies only to 2-D position space, which is spread across processors as a set of rectangular regions, each point accompanied by the corresponding 3V, ie. 3-D velocity space. Spectrally accurate convergence is demonstrated for periodic boundary conditions, using up to 32×48 cores on the MARCONI supercomputer.

5 Summary

This report has laid out initial work toward implementing a subset of the HERMES-3 equations in NEKTAR++. Some understanding of the basic physics of the blob2d example has been gained and some possible benchmark tests have been outlined. There are some difficulties in understanding the units used by the core code in HERMES-3 and other uncertainty comes in the form of the numerical diffusivities used in that code.

A simple continuation method was shown with the result that higher values of Rayleigh number can be accessed by FIREDRAKE convection simulations.

Grantee work over the year was summarized in Section 4.

Acknowledgement

ET acknowledges assistance with HERMES-3 from John Omotani of UKAEA, and Chris Ridgers and Mike Kryjak of the University of York, assistance with NEKTAR++ from the user community via the mailing list; also assistance with FIREDRAKE from Colin Cotter of Imperial College London and the FIREDRAKE online user community. ET also thanks Wayne Arter for pointing out the analogy between the plasma blob examples and fluid thermal convection.

The support of the UK Meteorological Office and Strategic Priorities Fund is acknowledged.

A A study of the Vlasov-Poisson equation system

The Vlasov-Poisson system of equations in $1+1$ dimensions (one space, one velocity), also referred to as $1d1v$, is the simplest example of a meaningful kinetic problem; technically this is a special case of the collisionless Boltzmann equation. Here a method of finding analytic solutions is outlined and demonstrated in simple cases.

The system is not a typical system of partial differential equations (Cauchy problem) because it contains explicit quadratures over one of the dimensions. It will be seen that the kinetic problem differs significantly from the fluid case and this provides (albeit limited) guidance for what to expect when exchanging the classical fluid picture of a plasma to a kinetic one (the full $3+3$-dimensional case involves seven independent variables and so cannot easily be solved, hence the focus on a toy example here). The problems discussed here represent collisionless systems where the self-interactions of the matter are of the collective, long-range nature (the matter density gives rise to a collective electrostatic field).

A.1 Example of an analytic equilibrium solution

The Vlasov-Poisson system is, for particles in a neutralizing background,

$$\begin{aligned} \frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + E(x)\frac{\partial f}{\partial v} &= 0; \\ \frac{d^2\Phi(x)}{dx^2} = \omega_P^2\left(\frac{1}{v_N}\int_{-\infty}^{+\infty} f dv - 1\right); \\ E &= -\frac{d\Phi}{dx}; \end{aligned} \tag{30}$$

where the constant v_N is chosen such that the domain contains no net charge. The constant ω_p is the plasma frequency, which specifies the strength of the coupling and is the only parameter in the problem (excepting initial data). Physically this translates into $\propto e\sqrt{n}$ for electron number density n for spatially-homogeneous solutions. The function f is the distribution function (DF) for the charged particles, ρ is the charge density, and Φ is the electrostatic potential.

It is clear that any spatially-homogeneous solution $f(v)$ and $E=0$ is a time-independent solution; the important question of the stability of such equilibria is deferred until later.

To find stationary solutions, let $f(x,v) = f(\mathcal{H})$ where

$$\mathcal{H} = \frac{1}{2}v^2 + \Phi(x). \tag{31}$$

This automatically reduces the Vlasov equation to $\partial f/\partial t = 0$ and then the associated potential is evaluated by inverting the nonlinear Poisson equation.

Note that $\mathcal{H}$ is just the Hamiltonian for a unit mass particle moving in a potential $\Phi(x)$ ie. a classical oscillator. The validity of the solution is easily shown by change of variable; more generally

the fact that a function of a conserved quantity solves the steady-state equations is known as Jeans' theorem. It is known from gravitational potential theory (eg. [20]) that this enables a range of equilibrium solutions to be constructed and studied.

The remaining part of the problem is the *linear* equation for the charge density

$$\rho = -\omega_P^2 \left(\int f \frac{dv}{v_N} - 1 \right) \tag{32}$$

Let us seek a solution where there is a maximum value of permitted energy ie. an energy scale is added into the problem. Thus define the constant Φ_0 to be this maximum and

$$\mathcal{E} \equiv \Phi_0 - \Phi - \frac{1}{2} v^2 \tag{33}$$

(note this automatically satisfies the Vlasov equation by the reasoning presented earlier) so that $\mathcal{E} > 0$ represents allowed values of v ie. $f > 0$ for $\mathcal{E} > 0$ and $f = 0$ for $\mathcal{E} \leq 0$. Further define $\Psi \equiv \Phi_0 - \Phi$.

Now one has

$$v_N \left(1 - \frac{\rho(x)}{\omega_P^2} \right) = \int_{f>0} f dv = 2 \int_0^\Psi \frac{f(\mathcal{E}) d\mathcal{E}}{\sqrt{2(\Psi - \mathcal{E})}}. \tag{34}$$

This is clearly of the form

$$g(\Psi) = \int f dv = \int_0^\Psi \frac{f(\mathcal{E}) d\mathcal{E}}{\sqrt{(\Psi - \mathcal{E})}} \tag{35}$$

and it is known that this equation has a unique solution for f given by the Abel integral formula[1]:

$$f(\mathcal{E}) = \frac{1}{\pi} \frac{\mathrm{d}}{\mathrm{d}\mathcal{E}} \int_0^\mathcal{E} \frac{g(t) dt}{\sqrt{\mathcal{E} - t}}. \tag{36}$$

Now ρ can be expressed in terms of Ψ using the Poisson equation (note change of sign)

$$\frac{d^2 \Psi}{dx^2} = \rho \tag{37}$$

so the general Abel inversion is

$$v_N \left(1 - \frac{1}{\omega_P^2} \frac{d^2 \Psi}{dx^2} \right) = \int_{f>0} f dv = 2 \int_0^\Psi \frac{f(\mathcal{E}) d\mathcal{E}}{\sqrt{2(\Psi - \mathcal{E})}}. \tag{38}$$

[1] which is a special case of the Radon transform, which is presumably the technique of choice for more general, higher-dimensional cases

A spatially-periodic system will be considered from this point. It is clear that the charge density function cannot have a nonzero $n = 0$ Fourier component, because there must be zero net charge. However, the potential $\Phi(x)$ may have a constant component as this is compatible with periodicity and it does not affect the dynamics.

Specialising to eg. $\rho = 4\pi^2 \cos 2\pi x$ on $[0, 1]$ one has $\Psi = \Phi_0 - \cos 2\pi x$ and $\rho = 4\pi^2 (\Phi_0 - \Psi)$.

(It is worth mentioning here that this is the main trick - one could also specify ρ implicitly by using something like $\frac{d^2\Phi}{dx^2} = -k^2 \sin \Phi$.)

Hence

$$\frac{v_N}{\sqrt{2}} \left(1 + \frac{4\pi^2}{\omega_P^2} (\Psi - \Phi_0) \right) = \int_{f>0} f dv = 2 \int_0^{\Psi} \frac{f(\mathcal{E}) d\mathcal{E}}{\sqrt{(\Psi - \mathcal{E})}}. \tag{39}$$

Applying Abel's formula gives

$$f(\mathcal{E}) = \frac{v_N}{\sqrt{2}\pi} \frac{\mathrm{d}}{\mathrm{d}\mathcal{E}} \int_0^{\mathcal{E}} \frac{\left(1 - \frac{4\pi^2\Phi_0}{\omega_P^2} + \frac{4\pi^2}{\omega_P^2} t \right) dt}{\sqrt{\mathcal{E} - t}}. \tag{40}$$

giving the DF as, for positive $\mathcal{E}$ (zero otherwise)

$$f(\mathcal{E}) = \frac{\mathrm{d}}{\mathrm{d}\mathcal{E}} \frac{\sqrt{2\mathcal{E}} v_N}{\pi} \left(1 - \frac{4\pi^2\Phi_0}{\omega_P^2} + \frac{8\pi^2\mathcal{E}}{3\omega_P^2} \right) \tag{41}$$

ie.

$$f(x, v) = \frac{v_N}{\sqrt{2}\pi} \left(\frac{1 - \frac{4\pi^2\Phi_0}{\omega_P^2}}{\sqrt{\Phi_0 - \cos 2\pi x - \frac{1}{2}v^2}} + \frac{8\pi^2 \sqrt{\Phi_0 - \cos 2\pi x - \frac{1}{2}v^2}}{\omega_P^2} \right) \tag{42}$$

wh. $\mathcal{E} \equiv \Phi_0 - \Phi - \frac{1}{2}v^2 = \Phi_0 - \cos 2\pi x - \frac{1}{2}v^2$. Here Φ_0 is a maximum energy level (free parameter).

Note there is a singularity as $\mathcal{E} \to 0$ albeit an integrable one (cf. $\int_0^1 \frac{dx}{\sqrt{1-x^2}} = \frac{\pi}{2}$). The case where $\Phi_0 = \frac{\omega_P^2}{4\pi^2}$ is special in that it avoids this singularity.

It is easy to substitute this solution back into the charge density equation to verify its correctness (as part of this, Φ_0 cancels out once the integration is done as it must as it does not appear in the charge density equation).

Note that $\omega_P \to \infty$ gives back the limit where the charge density does not depend on position (see this by evaluating the integral of f over velocity).

A.2 Summary

The Vlasov-Poisson system is, for particles in a neutralizing background,

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + E(x)\frac{\partial f}{\partial v} = 0;$$
$$\frac{d^2\Phi(x)}{dx^2} = \omega_P^2\left(\frac{1}{v_N}\int_{-\infty}^{+\infty} f dv - 1\right);$$
$$E = -\frac{d\Phi}{dx}. \tag{43}$$
$$\tag{44}$$

A particular example of a general steady-state solution is

$$\begin{aligned} f(x,v) &= \frac{v_N}{\sqrt{2}\pi}\left(\frac{1-\frac{4\pi^2\Phi_0}{\omega_P^2}}{\sqrt{\Phi_0 - \cos 2\pi x - \frac{1}{2}v^2}} + \frac{8\pi^2\sqrt{\Phi_0 - \cos 2\pi x - \frac{1}{2}v^2}}{\omega_P^2}\right) \\ \rho(x) &= -\omega_P^2\left(\int f\frac{dv}{v_N} - 1\right) = 4\pi^2\cos 2\pi x; \\ \frac{d^2\Phi}{dx^2} &= -\rho. \end{aligned} \tag{45}$$

This model contains only trapped particles if $\frac{\omega_P}{8\pi^2} + \frac{\Phi_0}{2} \leq 1$ (recall $\mathcal{E} = \Phi_0 - \Phi - \frac{1}{2}v^2$) but otherwise contains free particles. The non-singular case is on the limit of trapped particles and this limit is the only non-singular case, for which

$$f(x,v) = \frac{4\sqrt{2}\pi v_N}{\omega_P^2}\sqrt{\frac{\omega_P^2}{4\pi^2} - \cos 2\pi x - \frac{1}{2}v^2} \tag{46}$$

It is curious that this solution does not resemble more closely the states found by evolving unstable initial data as reported in [21]; those solutions seemed to be characterized by the formation of phase-space holes (probably present due to the incompressibility property of phase space), though the author is uncertain whether those states were actually time-independent.

This solution is unlikely to be a good candidate for simulation using spectral / hp methods due to the discontinuity in the derivative where the density terminates. It is also not known whether the solution is stable (the stability analysis is more involved than the examples exhibited in the next subsection, because the spatial variation means that a spatial perturbation is not a Fourier mode).

These type of states appear stationary but it is important to keep in mind that the model used here was collisionless; collisional systems need to be invariant under the action of particle collisions, which generally leads to Maxwellian-like velocity distributions.

A.3 Stability of equilibrium solutions

The stability of equilibrium solutions is of interest as it can provide tests of the time-evolution in a linear regime (in addition to answering the obvious question of whether a given solution is likely to

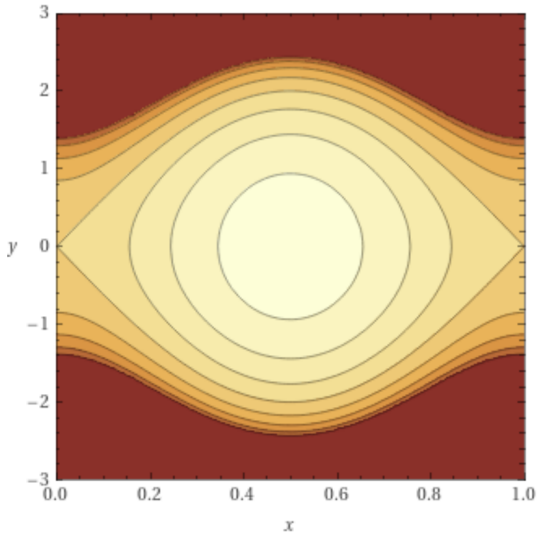


Figure 18: Contour plot of one x-period of the solution for $\frac{\omega_P^2}{8\pi^2} = 1$ and $\Phi_0 = 2$. Note this is a special choice of parameters for which the integrable singularity in f cancels away.

persist under time-evolution).

It is clear that any spatially-homogeneous solution $f(v)$ and $E = 0$ is a time-independent solution to Equation (30) and one consideration is to check its stability by computing the dispersion relation for a linear perturbation (the following standard formula applies for an x-independent background and a harmonic perturbation with frequency ω and wavenumber k - the latter of which is quantized if on a periodic spatial domain)

$$1 = \omega_p^2 \int_{-\infty}^{+\infty} \frac{f(v)dv}{(\omega - kv)^2}. \tag{47}$$

This comes from linearizing the system Equation (30) with eg. $f = f_0 + \epsilon f_1$ about the zero-electric-field, x-uniform initial distribution; the equation before integrating by parts is

$$1 = -\frac{\omega_P^2}{k} \int_{-\infty}^{\infty} \frac{df_0(u)}{du} \frac{du}{\omega - u} \tag{48}$$

in which $u = kv$ - note the integral is clearly the Hilbert transform of $\frac{df_0(u)}{du}$.

Note that the integrals over v mean that this is not a usual linear eigenvalue problem.

In the following, the stability of several interesting physical systems will be discussed. The analysis focusses on the two-stream instability problems of the sort investigated using a NEKTAR++ implementation in continuum kinetics in [21] and a *NESO-Particles* implementation using particle-in-cell in [22].

A.3.1 Gaussian beams

A stability analysis was done for the case of two counterpropagating Gaussian beams in [21]. There the initial f_0 is the bi-Gaussian (normed so that the whole system integrates over v to a density of 1) - for which total energy is $v_0^2 + \sigma^2$ (think of beam mean kinetic energy plus thermal energy and note the simple addition of these is due to Gaussianity) -

$$f_0(v) = \frac{1}{\sqrt{8\pi\sigma^2}} \left(e^{-\frac{(x-v_0)^2}{2\sigma^2}} + e^{-\frac{(x+v_0)^2}{2\sigma^2}} \right) \tag{49}$$

for which the dispersion integral can be done analytically to

$$1 = -\frac{\omega_P^2}{2\sigma^2 k^2} \left(2 - \sqrt{\pi} A e^{-A^2} \left(\mathrm{erfi}(A) - i \right) - \sqrt{\pi} A' e^{-A'^2} \left(\mathrm{erfi}(A') - i \right) \right) \tag{50}$$

with $A \equiv \frac{\omega - k v_0}{\sqrt{2}\sigma k}$, $A' \equiv \frac{\omega + k v_0}{\sqrt{2}\sigma k}$.

Note using a single Gaussian, or $v_0 = 0$ in the above, gives, in the small-k limit, the dispersion relation for a warm-plasma Langmuir wave $\omega^2 = \omega_P^2 + 3k^2\sigma^2$ using the series expansion of the Dawson function. It is also possible to show that for $v_0 = 0$ the system is stable (this is a Maxwellian profile in velocity space): as usual the boundary of the stable region is given by setting $\omega = 0$ in the dispersion relation. Now choosing the DF as

$$f(v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{v^2}{2\sigma^2}} \tag{51}$$

the Hilbert transform is trivial in the case $\omega = 0$, giving the criterion $\omega_P^2 = -k^2\sigma^2$. Since there are no solutions to this equation for any positive ω_P, the case $v_0 = 0$ is always stable.

It is known that there are pure imaginary solutions $\omega = i\gamma$ to Equation (50) representing purely growing or decaying modes, though actually plotting the dispersion $\omega(k)$ is challenging due to the stiffness of the resulting Newton solver scheme. One interesting feature is the boundary between stable initial data and unstable, which is found by inspecting where the sign of $\mathrm{Im}(\omega)$ changes. With ω purely imaginary over the range of interest, setting $\omega = 0$ in the above gives the stability frontier (Fig.19)

$$\frac{\omega_P}{k\sigma} = \frac{1}{\sqrt{2xD_+(x) - 1}} \tag{52}$$

where $x \equiv \frac{v_0}{\sqrt{2}\sigma} \equiv \sqrt{\frac{m_e v_0^2}{2k_B T}}$ and the Dawson function is $D_+(x) \equiv \frac{\sqrt{\pi}}{2} e^{-x^2} \mathrm{erfi}(x)$. The initial data is stable if ω_P is below the value given by the RHS. (Note the cold-beam case can be obtained from the $\sigma \to 0$ limit, stable for $\omega_P < k v_0$.)

This tells that the system is always stable for x below the zero at $2xD_+(x) = 1 \to x \approx 0.92414$ (where $2xD_+(x) < 1$ so the quantity in the radical is negative) and that above this there is a stability boundary where the system is stable for small enough values of the normalized coupling

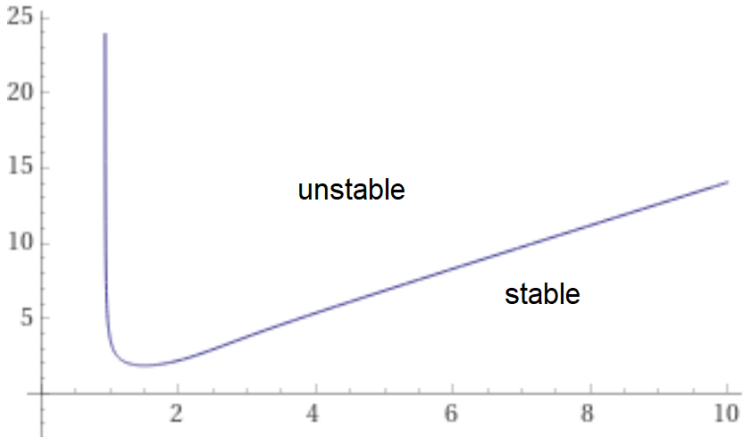


Figure 19: Stability frontier ('phase diagram') for the Gaussian initial condition two-stream instability: vertical axis is $\frac{\omega_P}{k\sigma}$ and horizontal is $\frac{v_0}{\sqrt{2}\sigma}$. Points below the curve represent stable initial conditions and overall scale is set by the Gaussian width, physically the temperature.

$\frac{\omega_P}{k\sigma}$. This boundary slopes upwards ie. for large enough normalized initial velocities the system is more likely to be stable. The picture obtained here is consistent with the expectation for the $\omega_P \to 0$ limit, in which any initial data must be stable as there is no dynamics. This formula also predicts evenly-spaced instability for modes of increasing integer wave number ie. the $n = 2$ mode requires twice the ω_P value to become unstable as does the $n = 1$ mode - hence it is apparent that the instability is a associated to long wavelengths (the latter is reminiscent of another kinetic-related linear instability, the Jeans instability).

The above analysis gives good agreement for the stability bound of the system when simulated in NEKTAR++, predicting instability when $\omega_P > 2.9223$ in that case (see the x-axis intercept in Fig.21).

The stability is here seen to be conditional on the value of the coupling strength Ω_P. Readers familiar with plasma physics might ask how this can be consistent with a Penrose's[2] criterion [23], which involves only the initial data (in the formula, u_0 is the location of a local minimum of $f(u)$, and satisfaction of the criterion is a necessary condition for the system to be unstable).

$$\int_{-\infty}^{\infty} \frac{f_0(u) - f(u_0)}{(u - u_0)^2}\, du > 0. \tag{53}$$

The answer is that the Penrose criterion applies to the non-periodic case in which there is no long-wavelength cut-off and in that case everything with a value of x above the zero at $2xD_+(x) = 1 \to x \approx 0.92414$ is unstable (indeed this condition seems to come out of doing the integral in the Penrose formula). The non-periodic case can also be accessed by plotting the graph for the periodic case and then rescaling the y axis by letting $k \to 0$, the stability curve will collapse to the x axis leaving the unstable region only for $x > 0.92414$.

This system may be viewed a sort of 'phase diagram' problem in the x, $y \equiv \frac{\omega_P}{k\sigma}$ plane. Note that

[2]Oliver Penrose, theoretical physicist and brother of the more popularly-known Roger

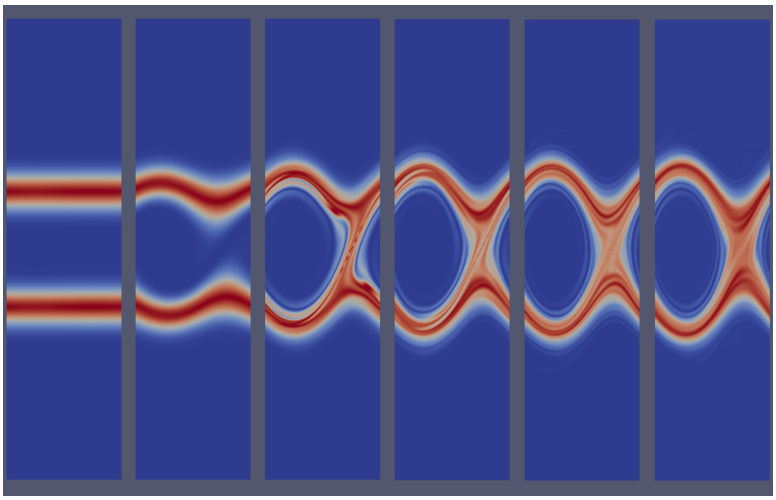


Figure 20: Time-evolution of the two-stream instability as a sequence of frames, taken from a NEKTAR++ simulation.

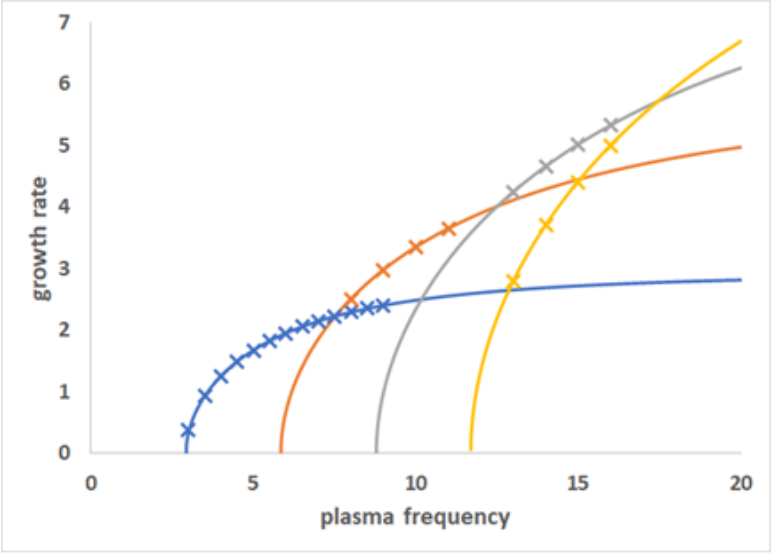


Figure 21: Dispersion relation for the two-stream instability for various values of k obtained from NEKTAR++ code (crosses) compared to the analytic solution in Equation (50) (curves). As evident from Eq.52 the ascending spatial harmonics are evenly spaced in ω_P with spacing of 2.9223, which is the value of the stability frontier ω_P for $\sigma = \frac{1}{5}$ and $v_0 = 1$ as used in the NEKTAR++ simulation.

the stability frontier has two asymptotics, $x = 0.92414$ and $y = \sqrt{2}x$.

The stability frontier and its two asymptotics are superficially similar to the phase diagram for the Lorenz model exhibited in [24] in which the behaviour of the Lorenz oscillator as a function of the usual Rayleigh and Prandtl numbers is studied; that system also exhibits an L-shaped stability frontier Figure 22. Note that the Rayleigh number corresponds to y and the Prandtl to x and that the axes are interchanged compared to Fig.19 ie. the coupling ω_P is analogous to the Rayleigh number and the velocity offset v_0 the Prandtl number (all scaled to the system temperature in the Vlasov-Poisson case).

A.3.2 Delta-function (ie. cold) beams

This case uses non-differentiable functions so it is not straightforward to apply the Penrose criterion. It is sometimes referred to as beams of cold plasma as it corresponds to zero-width Gaussian beams.

One has $f_0(v) = \frac{1}{2}\left(\delta(v - v_0) + \delta(v + v_0)\right)$. The dispersion relation is

$$z^2 = \frac{1}{2}\left(1 + 2y^2 \pm \sqrt{1 + 8y^2}\right) \tag{54}$$

wh. $z \equiv \frac{\omega}{\omega_P}$, $y \equiv \frac{kv_0}{\omega_P}$. I think the negative sign for the radical to be the correct choice. Considering stability bounds, setting $z = 0$ one finds a zero at $y = 1$ ie. $\omega_P = kv_0$ and it seems the system is stable for $\omega_P < kv_0$. Clearly for $k = 0$ (continuum case) the system is always unstable.

A.3.3 Top hat function beams

Normalized to unity in total, two top hats centred on $\pm v_0$ and both width = σ. Again, it is not straightforward to apply the Penrose criterion.

$$z^2 = \frac{1}{2}\left(1 + 2y^2 + 2s^2 - \sqrt{1 + 8y^2 + 16s^2y^2}\right) \tag{55}$$

with $s \equiv \frac{k\sigma}{\omega_P}$.

One obtains the fact that this case is stable up to $\omega_P^2 = k^2(v_0^2 - \sigma^2)$ (solve quadratic and take positive root to ensure recovery of $s = 0$ limit) so again always unstable in the continuum and the width actually makes the system *more* unstable in the sense that if the top hats touch or overlap the system is always unstable even if periodic (weirdly, this means a single top hat centred on zero is always unstable and this is a function with no minimum!). The stability frontier can be expressed in the form

$$\frac{\omega_P}{kv_0} = \frac{\sqrt{x^2 - 1}}{x} \tag{56}$$

wh. $x \equiv \frac{v_0}{\sigma}$.

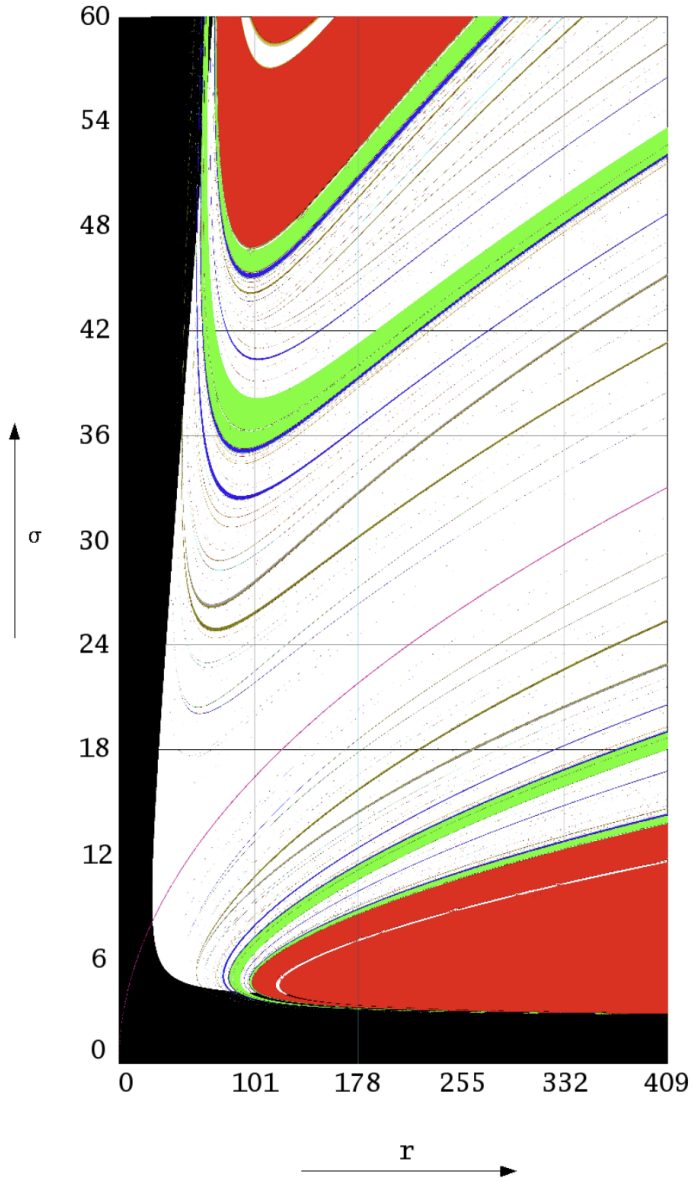


Figure 22: Phase diagram of the Lorenz 63 model in $(\mathrm{Pr}, \mathrm{Ra})$ space (in the figure r is Rayleigh number and σ Prandtl number). The black regions correspond to values in parameter space where the system tends to a stable fixed point (ie. there is no chaotic behaviour). Figure reproduced from [24].

A.3.4 Cauchy (or Lorentzian, or Breit-Wigner) beams

This is a bit questionable as the second moment of the distribution is not finite (suggesting unbounded temperature); the distribution is

$$f_0(v) = \frac{s}{2\pi}\left(\frac{1}{(v-v_0)^2+s^2} + \frac{1}{(v+v_0)^2+s^2}\right) \tag{57}$$

and in this case the integration can be done thanks to a formula obtained from the useful *WolframAlpha* website [25]

$$\int_{-\infty}^{\infty} \frac{dv}{(v^2+s^2)(\omega-kv)^2} = -\frac{\pi}{s(ks-i\omega)^2} \tag{58}$$

and doing a linear transformation on v. The resulting dispersion relation, with ω set zero to find the stability contour, gives

$$\frac{\omega_P}{kv_0} = \frac{x^2+1}{\sqrt{x^2-1}} \tag{59}$$

where as usual $x \equiv \frac{v_0}{s}$.

The asymptotics are $y = 1$ and $y = x$, trivial.

This stability contour is very much like that obtained in the Gaussian case, and indeed in the $k = 0$ limit (non-periodic) the system is stable for $s > v_0$. The latter concurs with the non-periodic Penrose formula analysis done at [23]. This case shows that Gaussianity is not required for stable solutions and indeed the v_0 limit represents a single Cauchy distribution centred on zero - so the dynamics here, unlike the collisional case, admit multiple stable solutions and not just Maxwellians.

The full dispersion relation for this case is obtained by solving the equation

$$\frac{2}{\bar{v}^2} = \frac{1}{(iy+z)^2} + \frac{1}{(i\bar{y}+z)^2} \tag{60}$$

wh. $z \equiv \frac{\omega}{k}$, $y \equiv s + iv_0$ and $\bar{v} \equiv \frac{\omega_P}{k}$. The solution is

$$\frac{\omega}{ks} = \frac{i}{2}\left(-2 \pm \sqrt{\mp 2x^2\sqrt{1+\frac{8y^2}{x^2}} - 4y^2 - 2x^2}\right) \tag{61}$$

which can be plotted (but is not very interesting beyond observing the existence of the asymptotics). Doubtless if some QoI associated to the full nonlinear dynamics were used, the plot would be richer (as is the Lorenz case in [24]). Note that this dispersion relation is easier to work with numerically than the one for the Gaussian case Eq.Equation (50) as it does not involve multiplication of very large and very small factors (as well as not using higher functions).

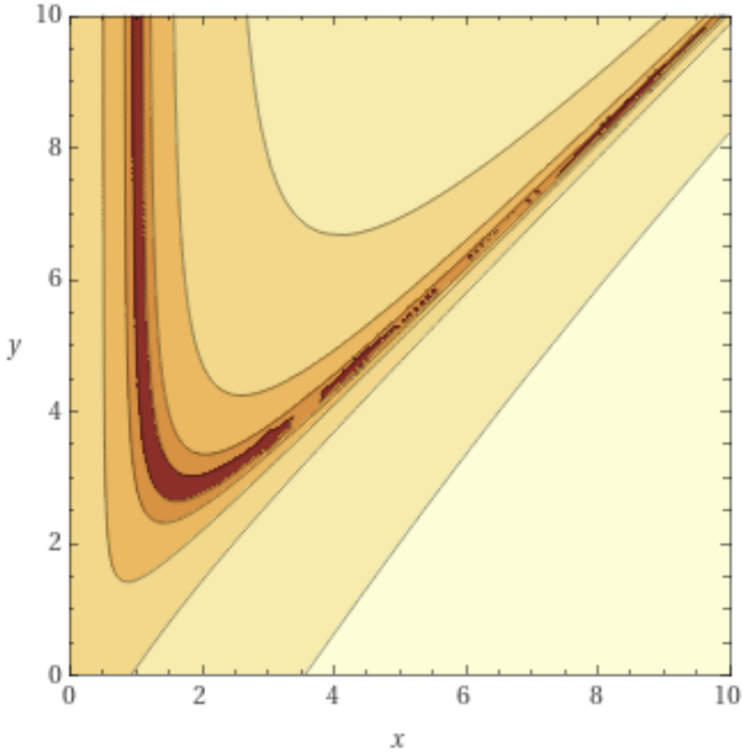


Figure 23: Dispersion plot for Cauchy case with here $x \equiv \frac{v_0}{s}$ and $y \equiv \frac{\omega_P}{ks}$. The natural logarithm has been taken to highlight the stability contour given by Eq.59. Generated using *WolframAlpha* with the input `plot log(-2+sqrt(2x ^2 sqrt(1+8y^2/x^2)-4y^2-2x^2)) from 0 to 10`. Note similarity to Fig.19.

A.3.5 A simple comparison of kinetic and fluid two-stream problems

One can ask whether there is a fluid analogue of the two-stream instability. Here, a particular fluid set-up is considered where the only mutual interaction is via an electrostatic force and there are two fluids of negative charge (with charge neutrality enforced with a non-dynamic positive charge background) forming one-dimensional beams with bulk velocity $\pm\bar{v}$ (of course, there is Maxwellian thermal velocity on scales much below the hydrodynamic scale). The fluids must be compressible since the dynamics of a one-dimensional incompressible fluid are trivial.

The equations of motion are, noting negative charge for both fluids, and the neutralization term being implicit and constant in space and time

$$\begin{aligned}
\frac{\partial \rho_1}{\partial t} + (\rho_1 v_1)' &= 0; && (62)\\
\frac{\partial v_1}{\partial t} + v_1 v_1' &= -\frac{1}{\rho_1} p_1' + \frac{q}{\rho_1}\Phi'; && (63)\\
\frac{\partial \rho_2}{\partial t} + (\rho_2 v_2)' &= 0; && (64)\\
\frac{\partial v_2}{\partial t} + v_2 v_2' &= -\frac{1}{\rho_2} p_2' + \frac{q}{\rho_2}\Phi'; && (65)\\
\Phi'' &= \frac{q}{\epsilon_0}(\rho_2 + \rho_1). && (66)
\end{aligned}$$

Now note the sound speed is given by $p_1 = v_s^2 \rho_1$ and is the same for both fluids.

Now the background state has $\rho_1 = \rho_2 = \bar{\rho}$ and $v_1 = -v_2 = \bar{v}$ as constants and zero electric potential given the neutralization.

Doing a linear expansion about the background state one has, having time-diffentiated the continuity equation and space-differentiated the momentum equation

$$\begin{aligned}
\left(\omega^2 + \bar{v}k\omega - v_s^2 k^2 - \omega_P^2\right)\rho_1 - \omega_P^2 \rho_2 &= 0 && (67)\\
-\omega_P^2 \rho_1 + \left(\omega^2 - \bar{v}k\omega - v_s^2 k^2 - \omega_P^2\right)\rho_2 &= 0 && (68)
\end{aligned}$$

where $\omega_P^2 \equiv \frac{\bar{\rho} q^2}{\epsilon_0}$.

The dispersion relation is obtained from the usual determinant. The stability frontier is obtained by solving in the $\omega = 0$ case and one obtains the simple stability criterion

$$k \geq 0. \qquad (69)$$

This means that the charged fluid two-stream case is always stable for any value of the coupling strength. There is no analogue of the Jeans wavenumber that occurs in the gravitational case - this is because the two electrostatically-charged fluids here experience a mutual repulsion, not an

attraction. The electrostatic two-stream instability is seen to be a purely kinetic effect (or, at least, the instability requires more terms than are retained in the model of the system by compressible Navier-Stokes). The fluid result can be compared with the delta-function kinetic case in which the stability criterion is

$$\omega_P < kv_0. \tag{70}$$

This is always unstable in infinite space (but it may be stable in a periodic system because there is in that case a minimum permitted value for k).

The full dispersion relation is easily solved to

$$y^2 = \left(z^2 + \frac{1}{2}\right) x^2 + 1 - \sqrt{\left(z^2 + \frac{1}{4}\right) x^4 + x^2 + 1} \tag{71}$$

where $y \equiv \frac{\omega}{\omega_P}$, $x \equiv \frac{\bar{v}k}{\omega_P}$, and $z \equiv \frac{v_s}{\bar{v}}$. This is the continuum version of the kinetic theory formula Equation (50).

Note large z is the subsonic limit in which is recovered sound waves, $\omega = v_s k$. In the supersonic $z = 0$ limit there are no waves ($\omega = 0$).

Note at least in the static limit, stability is expected since the system will act to neutralize a local perturbation in charge density and not to amplify such a perturbation.

B *Units in vertical convection*

A sketch of the derivation of the equations of Rayleigh-Bénard convection using the Boussinesg approximation is presented in eg. Tritton [26, § 14 Appendix], with result

$$\begin{aligned} \frac{\partial u}{\partial t} + u \cdot \nabla u &= -\frac{1}{\rho_0}\nabla p + \nu \nabla^2 u + g\alpha(T - T_0)\hat{y}; \\ \frac{\partial T}{\partial t} + u \cdot \nabla T &= \kappa \nabla^2 T; \\ \nabla \cdot u &= 0. \end{aligned} \tag{72}$$

where ρ_0 is the fluid mass density at a reference temperature T_0, so that ν is then the viscous diffusivity. Coefficient κ is thermal diffusivity, both with dimension $L_s^2 t_s^{-1}$, and note that $p' = p/\rho_0$ has dimension $(L_s/t_s)^2$.

Rayleigh-Bénard convection is driven by a temperature difference between opposite boundaries, and the lengthscale L_s is taken to be the distance between these boundaries, usually the layer depth. Even so, there are several ways to non-dimensionalize the equations; understanding of these is necessary to convert between the scalings used in the literature. The resulting systems are, of course, physically equivalent.

B.1 Scaling to Rayleigh and Prandtl number using thermal diffusivity timescale

Rescale time t as $t = t_0\tilde{t}$ using $t_0 = \frac{L_s^2}{\kappa}$, $u = u_0\tilde{u}$, and $T = T_0 + \Delta T\tilde{T}$ where the tilded quantities are dimensionless and $u_0 = L_s/t_0$.

The equations become in the dimensionless quantities, and dropping the tildes,

$$\begin{aligned}
\frac{1}{\mathrm{Pr}}\left(\frac{\partial u}{\partial t} + u\cdot\nabla u\right) &= -\nabla p' + \nabla^2 u + \mathrm{Ra}\, T\hat{y}; && (73)\\
\frac{\partial T}{\partial t} + u\cdot\nabla T &= \nabla^2 T; && (74)\\
\nabla\cdot u &= 0. && (75)
\end{aligned}$$

The dimensionless Rayleigh and Prandtl numbers are respectively

$$\begin{aligned}
Pr &\equiv \frac{\nu}{\kappa},\\
Ra &\equiv \frac{g\alpha\Delta T L_s^3}{\kappa\nu}. && (76)
\end{aligned}$$

This formulation is widely used eg. ref [27] and notably in the classic 1960s studies by Elder [28] and in the papers by Winters [29].

B.2 Scaling to Rayleigh and Prandtl number using free-fall velocity

The free-fall velocity is

$$u_0 = \sqrt{gL_s\alpha\Delta T} \qquad (77)$$

and the rescaling is $t = t_0\tilde{t}$, $u = u_0\tilde{u}$, and offset temperature $T - T_0 = \Delta T\tilde{T}$ where the tilded quantities are dimensionless and $t_0 = L_s/u_0$.

The equations become (as before, dropping the tildes)

$$\begin{aligned}
\frac{\partial u}{\partial t} + u\cdot\nabla u &= -\nabla p' + \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}}\nabla^2 u + T\hat{y}; && (78)\\
\frac{\partial T}{\partial t} + u\cdot\nabla T &= \frac{1}{\sqrt{\mathrm{Ra}\,\mathrm{Pr}}}\nabla^2 T; && (79)\\
\nabla\cdot u &= 0. && (80)
\end{aligned}$$

This formulation is used in Zucatti et al. [30] and Wang et al. [8] (and is useful eg. for converting the length of calculation needed for the system to relax to the quasi-steady state quoted in the appendix to the latter paper).

B.3 Scaling to Grashof number using viscosity

Rescale time t as $t = t_0\tilde{t}$ using $t_0 = \frac{L_s^2}{\nu}$, $u = u_0\tilde{u}$, and $T - T_0 = \Delta T\tilde{T}$ where the tilded quantities are dimensionless and $u_0 = L_s/t_0$.

The equations become, dropping the tildes, and with Prandtl number defined as above Equation (73)

$$\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) = -\nabla p' + \nabla^2 u + \mathrm{Gr}\, T\hat{y}; \tag{81}$$

$$\frac{\partial T}{\partial t} + u \cdot \nabla T = \frac{1}{\mathrm{Pr}}\nabla^2 T; \tag{82}$$

$$\nabla \cdot u = 0. \tag{83}$$

The Grashof number is

$$\mathrm{Gr} \equiv \frac{g\kappa\Delta T L_s^3}{\nu^2} \equiv \frac{\mathrm{Ra}}{\mathrm{Pr}}. \tag{84}$$

Two relevant cases are the $\mathrm{Pr} \to \infty$ limit in which there is zero thermal diffusivity, and $\mathrm{Pr} = 1$ where the diffusivities for momentum and for temperature are identical.

This formulation is used in the FIREDRAKE Rayleigh-Bénard convection example at [9].

B.4 Scaling with the Ad hoc Approach

Rescale time t by t_s, velocity by $u_s = L_s/t_s$ and offset temperature by T_s, where t_s, L_s and T_s may be chosen arbitrarily.

The equations become, dropping the tildes on the fields

$$\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) = -\nabla p' + \tilde{\nu}\nabla^2 u + \mathrm{Ra}_{ahc} T\hat{y}; \tag{85}$$

$$\frac{\partial T}{\partial t} + u \cdot \nabla T = \tilde{\kappa}\nabla^2 T; \tag{86}$$

$$\nabla \cdot u = 0. \tag{87}$$

The resulting Rayleigh number is

$$\mathrm{Ra}_{ahc} = \frac{g\alpha T_s t_s^2}{L_s} \times \frac{\Delta T}{T_s} \tag{88}$$

This formulation is particularly convenient for engineering problems, where the diffusivities will normally be significant functions of temperature that might directly introduced, and T_s may be

taken in units of Kelvin. Specifically $\tilde{\kappa} = \kappa t_s / L_s^2$ and $\tilde{\nu} = \nu t_s / L_s^2$ with equations

$$\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) = -\nabla p' + \nabla \cdot \tilde{\nu} \nabla u + \mathrm{Ra}_{ahc} T \hat{y}; \tag{89}$$

$$\frac{\partial T}{\partial t} + u \cdot \nabla T = \nabla \cdot \tilde{\kappa} \nabla T; \tag{90}$$

$$\nabla \cdot u = 0. \tag{91}$$

References

[1] W. Arter. Equations for ExCALIBUR / NEPTUNE proxyapps. Technical Report CD/EXCALIBUR-FMS/0021, UKAEA Project Neptune, 2023. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/ukaea_reports/CD-EXCALIBUR-FMS0021-1.26-M1.2.1.pdf.

[2] Hermes-3. https://hermes3.readthedocs.io/en/latest/examples.html#d-drift-plane. Accessed: March 2023.

[3] O.E. Garcia, N.H. Bian, and W. Fundamenski. Radial interchange motions of plasma filaments. *Physics of Plasmas* **13***, 082309 (2006)*, 2006.

[4] J.T. Omotani, F. Militello, L. Easy, and N.R. Walkden. The effects of shape and amplitude on the velocity of scrape-off layer filaments. *Plasma Physics and Controlled Fusion*, 58(1):014030, 2015.

[5] Nektar-Driftwave. https://github.com/ExCALIBUR-NEPTUNE/nektar-driftwave. Accessed: March 2023.

[6] Nektar-Driftplane. https://github.com/ExCALIBUR-NEPTUNE/nektar-driftplane. Accessed: March 2023.

[7] B. Dudson, P.A. Hill, E. Higgins, D. Dickinson, S. Wright, and D. Moxey. Task 2.2 BOUT++ 1D fluid solver with realistic boundary conditions : implementation. Technical Report 2047356-TN-07-1, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047356/TN-07-1.pdf.

[8] Q. Wang, H-R. Liu, R. Verzicco, O. Shishkina, and D. Lohse. Regime transitions in thermally driven high-Rayleigh number vertical convection. *J. Fluid Mech. (2021) Vol.197 A.6 doi:10.1017/jfm.2021.262*, 2021.

[9] Rayleigh-Benard convection. https://www.firedrakeproject.org/demos/rayleigh-benard.py.html. Accessed: March 2023.

[10] W.V.R Malkus. The heat transport and spectrum of thermal turbulence. *Proc. R. Soc. London A* **225** *(1161) 196-212*, 1954.

[11] M. Green, D. Moxey, C. Cantwell, Bin Liu, and S.J. Sherwin. D1.2: Augmentation of the NekMesh generator to provide quad-based meshes for 2D configurations. Technical Report 2053622-TN-03, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2053622/TN-03.pdf.

[12] Bin Liu, C. Cantwell, S.J. Sherwin, M. Green, and D. Moxey. D2.4: Evaluation of preconditioners of matrix-free operations in Nektar++ for anisotropic heat transport. Technical Report 2053622-TN-04, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2053622/TN-04.pdf.

[13] M. Barnes, F.I. Parra, M.R. Hardman, and J. Omotani. Numerical study of 1+1D drift kinetic models for parallel dynamics in the plasma edge. Technical Report 2047357-TN-10, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047357/TN-10.pdf.

[14] M.R. Hardman, J. Omotani, M. Barnes, and F.I. Parra. Numerical study of a reduced model coupling 2D+2V drift-kinetic ions and 2D+3V kinetic neutrals in a helical magnetic field with wall boundaries. Technical Report 2047357-TN-13-2, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047357/TN-13-2.pdf.

[15] M. Barnes, M.R. Hardman, J. Omotani, and F.I. Parra. Overview of the numerical issues and findings associated with the 1D and 2D drift kinetic models. Technical Report 2047357-TN-14-2, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047357/TN-14-2.pdf.

[16] M. Barnes, F. I. Parra, M. R. Hardman, and J. Omotani. Numerical study of 1+1D drift kinetic models for parallel dynamics in the plasma edge. Technical Report 2047357-TN-10, UKAEA Project Neptune, 2021. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047357/TN-10.pdf.

[17] M. Barnes, M.R. Hardman, J. Omotani, and F.I. Parra. Update on state of the art in edge modelling. Technical Report 2070839-TN-01, UKAEA Project Neptune, 2023. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2070839/TN-01.pdf.

[18] M.R. Hardman, J. Omotani, and M. Barnes. Implementing distributed-memory in the 2D-3V drift-kinetic edge code. Technical Report 2070839-TN-02, UKAEA Project Neptune, 2023. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2070839/TN-02.pdf.

[19] F. I. Parra, M. Barnes, and M.R. Hardman. Physics in the edge of fusion devices. Technical Report 2047357-TN-03-02, UKAEA Project Neptune, 2021. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047357/TN-03.pdf.

[20] J. Binney and S. Tremaine. *Galactic dynamics*. Princeton University Press, 2011.

[21] E. Threlfall and W. Saunders. Support high-dimensional procurement. Technical Report CD/EXCALIBUR-FMS/0066, UKAEA Project Neptune, 2022. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/ukaea_reports/CD-EXCALIBUR-FMS0066-M4.1.pdf.

[22] J. Cook, W. Saunders, and W. Arter. 1-D and 2-D particle models. Technical Report CD/EXCALIBUR-FMS/0070, UKAEA Project Neptune, 2022.

[23] Velocity-Space Instabilities. `https://farside.ph.utexas.edu/teaching/plasma/Plasma/node117.html`. Accessed: March 2023.

[24] H. Dullin, S. Schmidt, P.H. Richter, and S.K. Grossmann. Extended phase diagram of the Lorenz model. *International Journal of Bifurcation and Chaos Vol.17 No.09, pp.3013-3033*, 2007.

[25] WolframAlpha. `www.wolframalpha.com`. Accessed: September 2022.

[26] D.J. Tritton. *Physical Fluid Dynamics, 2nd Edition*. Clarendon Press, Oxford, 1988.

[27] N. Boullé, P.E. Farrell, and M.E. Rognes. Optimization of hopf bifurcation points, 2022.

[28] J.W. Elder. Numerical experiments with free convection in a vertical slot. *Journal of Fluid Mechanics*, 24:823–843, 1965.

[29] K.H. Winters. Hopf bifurcation in the double-glazing problem with conducting boundaries. *Transactions of the ASME*, 109:894, 1987.

[30] V. Zucatti, H.F.S. Lui, D.B. Pitz, and W.R. Wolf. Assessment of reduced-order modelling strategies for convective heat transfer. *Numerical Heat Transfer: Part A: Applications 77:7.*, pages 702–729, 2020.