UK Atomic Energy Authority

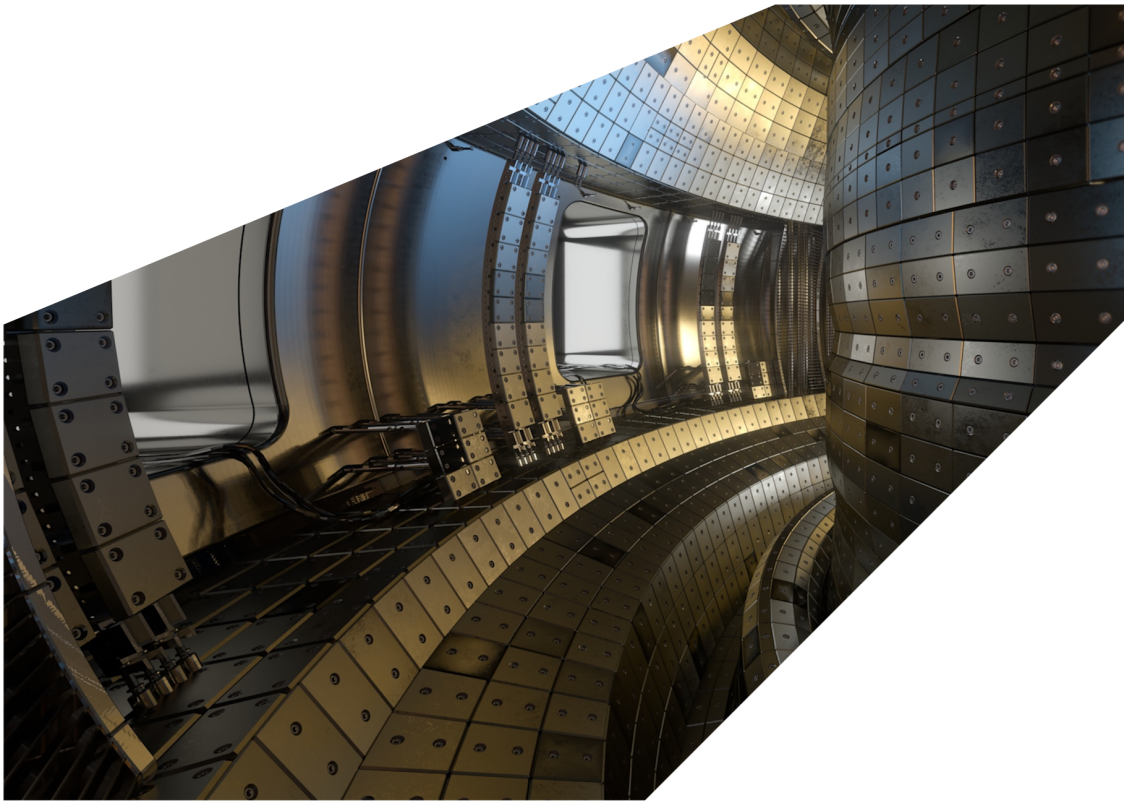
# ExCALIBUR

## Literature review for Call T/AW086/21: “Mathematical Support for Software Implementation”

## M7.1

**Abstract**

The report describes work for ExCALIBUR project NEPTUNE at Milestone 7.1. This is a literature review performed to support the Call T/AW086/21: “Mathematical Support for Software Implementation”.



**UKAEA REFERENCE AND APPROVAL SHEET**

| | | |
|---|---|---|
| | Client Reference: | |
| | UKAEA Reference: | CD/EXCALIBUR-FMS/0052 |
| | Issue: | 1.00 |
| | Date: | September 16, 2021 |
| Project Name: ExCALIBUR Fusion Modelling System | | |

| | Name and Department | Signature | Date |
|---|---|---|---|
| Prepared By: | Joseph Parker<br>Wayne Arter<br><br>BD | N/A<br>N/A | September 16, 2021<br>September 16, 2021 |
| Reviewed By: | Rob Akers<br><br>Advanced Computing Dept. Manager | | September 16, 2021 |
| Approved By: | Rob Akers<br><br>Advanced Computing Dept. Manager | | September 16, 2021 |

# 1 Introduction

Call T/AW086/21 “Mathematical Support for Software Implementation” is the second in a series of calls for mathematical support, following Call T/NA084/20 “Investigate matrix-preconditioning techniques”. This task focuses upon the suitability of available numerical algorithms (or the development of new algorithms) for Exascale-targeted plasma modelling.

The ideal numerical algorithms for future Exascale edge plasma codes would have preferably at least the following properties:

- P1 Accurate solution of hyperbolic problems.
- P2 Ability to deliver efficient and accurate solutions of corresponding elliptic problems.
- P3 Accurate modelling of highly anisotropic dynamics.
- P5 Accurate representation of velocity (phase) space.
- P6 Preservation of conservation properties of the underlying equations.
- P7 Scalability to likely Exascale architectures:
  - a interaction between models of different dimensionality,
  - b interaction between particle and fluid models,
  - c dynamic construction of surrogates.
- P8 Performance portability to allow rapid deployment upon emerging hardware.

It is unlikely that any algorithm will have all the above, and part of the call exercise will be to rank the importance of these properties.

Work under the previous NEPTUNE calls has provided

1. a survey of preconditioning methods, identifying suitable methods for Project NEPTUNE software, see ref [1, 2]
2. strong indication that time advance methods could benefit from use of larger timesteps without significant loss of accuracy
3. as yet, no identification of a satisfactory manner to treat extremely anisotropic transport of plasma
4. a study of particle codes showing the importance of using a control-variates approach [2], ie. of carefully handling of statistics, but
5. as yet, no identification of a satisfactory manner to coupling continuum (fluid) and particle models of plasma together in the different ways required by NEPTUNE [3].

It has also become evident that many arguments for Exascale suitability of a particular approach would be strengthened by mathematical analysis of the proposed algorithm. Indeed, many techniques used in modelling systems of high dimensionality modelling (dimension five or larger) involve random sampling techniques where a knowledge of stochastics is likely invaluable. In addition, certain cutting-edge algorithms such as "asymptotic-preserving" methods and Variable Stepsize-Variable Order timestepping require advanced mathematical skills to understand and thus to adapt for implementation in NEPTUNE software.

This report provides a literature review designed to inform the production of Call T/AW086/21 "Mathematical Support for Software Implementation". It contains two main sections concerning recent advances in algorithm development. Section 2.1 discusses methods used for the time advance of hyperbolic equations, while section 2.2 discusses the solution of elliptic equations. The work is summarized in Section 3.

# 2 Algorithms for NEPTUNE-relevant equation systems

Modelling the physical systems relevant to NEPTUNE will entail solving a partial differential equation system where hyperbolic equations are to be evolved in time. Evolving hyperbolic equations implicitly often requires the solution of a corresponding elliptic equation. Moreover, in relevant physical systems hyperbolic equations are often constrained by an elliptic equation which must be solved at every timestep (for example, in the Vlasov–Poisson system, the Poisson equation must be solved at every timestep to determine the electrostatic potential needed to advance the Vlasov equation). In this section, we discuss recent advances in algorithms for solving hyperbolic systems (in section 2.1) and elliptic systems (in section 2.2).

## 2.1 Hyperbolic systems

In this subsection, we discuss recent work on algorithms for the time-advance of hyperbolic systems. In particular we discuss variants of Implicit-Explicit (IMEX) time advance schemes which have additional favourable properties – here, schemes which have an increased order-of-accuracy from incorporating deferred corrections, and variants which are asymptotic preserving. We also discuss Variable Stepsize, Variable Order (VSVO) schemes.

### 2.1.1 IMEX Schemes

Multi-physics systems typically exhibit multiple timescales. In algorithms, the timescales for multi-physics systems are often separated into fast and slow timescales, where "slow" may refer to either a slow physical timescale or to a large computational cost of calculating the effect on a parallel machine. The canonical example of this approach is an Implicit-Explicit (IMEX) scheme. Such schemes seek to integrate the equation

$$\frac{\mathrm{d}u}{\mathrm{d}t} = R(x,u,t) = F(x,u,t) + G(x,u,t), \tag{1}$$

in time by decomposing the right-hand side $R$ into terms $F$ and $G$ which are responsible for fast and slow timescales respectively. Denoting the approximation to $u$ at time $t$ as $u_n = u(t_n)$ and approximating $\mathrm{d}u/\mathrm{d}t = (u_{n+1} - u_n)/\Delta t$, equation (1) may be written as

$$u_{n+1} = u_n + \Delta t\ F(x, u_{n+1}, t_{n+1}) + \Delta t\ G(x, u_n, t_n), \tag{2}$$

and then integrated in two steps

$$u^* = u_n + \Delta t\ G(x, u_n, t_n), \tag{3a}$$

$$u_{n+1} - \Delta t\ F(x, u_{n+1}, t_{n+1}) = u^*, \tag{3b}$$

where (3a) is an explicit method, and (3b) may be solved by an implicit scheme. Other choices of approximation to $\mathrm{d}u/\mathrm{d}t$ lead to different schemes, with IMEX schemes generally being classified as IMEX Runge–Kutta or IMEX multistep methods. There is also work on IMEX schemes for General Linear Methods (which are generalizations of Runge–Kutta and multistep methods) [4].

This approach allows large time steps to be taken in the implicit integration of $F$ (associated with the fast time scale), rather than forcing $\Delta t$ to limited by the typically-smaller stability limit of an explicit scheme. This approach is particularly fruitful when $G$ is nonlinear (and thus difficult to integrate implicitly) or expensive to compute (so that the repeated evaluations necessary for iterative methods are undesirable).

Variations on this fundamental IMEX idea is an area of ongoing research, with a typical approach being to incorporate desirable properties into IMEX schemes. In the following sections we discuss some examples of this: using deferred corrections to increase the order of schemes, ensuring schemes are asymptotic preserving, and combining IMEX with high-order multiderivative schemes.

**Deferred Correction Methods** One line of work combines IMEX schemes with Spectral Deferred Correction (SDC) [5] and Integral Deferred Correction (InDC) schemes [6]. Both SDC and InDC were originally developed in the context of solving ODEs [7, 8] and are based on the older Deferred Correction (DC) method [9, 10]. In Deferred Correction, one integrates the ODE for $N$ time steps with a $k$th-order method, and then interpolates the numerical solution with a (unique) $N$th order polynomial. Substituting this polynomial back into the original equation, one may derive an "error equation", an ODE for the difference between the true solution and the polynomial approximation. Solving this equation with the same $k$th order method yields corrections to add to the original numerical approximation to make it closer to the true solution. It may be shown that the corrected solution is $(2k)$th order accurate [9, 10], and therefore a higher-order method has been constructed automatically from a low-order method.

SDC and InDC follow this idea, except in these approaches the deferred correction is applied to the error equation formulated as a Picard integral equation rather than an ODE. This approach has been shown to have better stability and accuracy properties [7, 8]. The difference between SDC and InDC is in the choice of quadrature grid for solving the error equation: SDC uses Gauss, Lobatto or Radau grids as this improves stability and accuracy properties, while InDC uses a uniform grid as this ensures that increasing the order of the low-order method will propagate through to increase the order of the corrected scheme. This allows very high-order schemes to be constructed systematically using only low-order schemes [11].

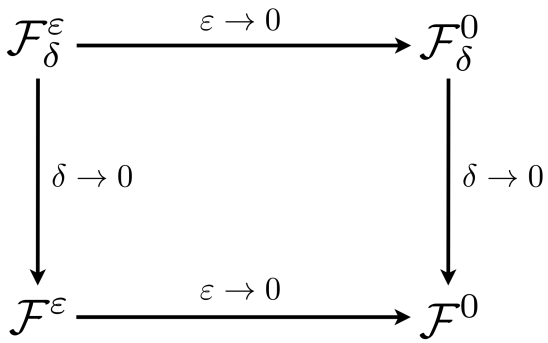



Figure 1: Illustration of the relationship between a microscopic model $\mathcal{F}^\varepsilon$, its macroscopic limit $\mathcal{F}^0$, and their discretizations $\mathcal{F}_\delta^\varepsilon$, $\mathcal{F}_\delta^0$, where $\varepsilon$ and $\delta$ are parameters characterising small physical and numerical scales respectively. The discretization scheme $\mathcal{F}_\delta^\varepsilon$ is said to be asymptotic preserving if $\mathcal{F}_\delta^0$ is a consistent and stable approximation to $\mathcal{F}^0$. Figure taken from [12].

**Asymptotic Preserving Schemes** In some models of multiscale physical systems, the scales which the model describes may be determined by a parameter. For example the Vlasov equation is a model for a plasma whose behaviour depends on the collision time (the typical time between particle interactions). When the collision time is large, the Vlasov equation is a kinetic equation for the plasma, encompassing microscopic scales in velocity space. However, when the collision time is small, the particle distribution function does not deviate significantly from a Maxwellian, and the solution only contains macroscopic velocity scales.

Discretizations of multiscale systems are not guaranteed to correctly capture the behaviour in the macroscopic limit, but numerical schemes may be derived to do so. Such schemes are called *asymptotic preserving*, and may be understood schematically as in Figure 1 (taken from [12]). Let $\mathcal{F}^\varepsilon$ be the multiscale model, with small scales characterised by the parameter $\varepsilon$, such that in the limit $\varepsilon \to 0$, we have the macroscopic model $\mathcal{F}^0$. Let $\mathcal{F}_\delta^\varepsilon$ be a discretization of $\mathcal{F}^\varepsilon$, where $\delta$ characterises small numerical scales, typically mesh spacing or timestep. Then in the macroscopic limit $\varepsilon \to 0$, we will also have a discretization of the macroscopic model, $\mathcal{F}_\delta^0$. If $\mathcal{F}_\delta^0$ is a consistent and stable approximation to $\mathcal{F}^0$, then the discretization scheme $\mathcal{F}_\delta^\varepsilon$ is said to be asymptotic preserving.

The value of asymptotic preserving schemes is that they allow the same model to be used to study both microscopic and macroscopic regimes simply by varying the parameter $\varepsilon$. Without asymptotic preserving schemes, one must either derive a separate model for the macroscopic scales, or obtain the macroscopic behaviour by solving the (typically much more expensive) microscopic system directly with large $\varepsilon$.[1] However, specific asymptotic preserving schemes must be derived for individual physical systems. Moreover, asymptotic preserving schemes are typically stiff due to the smallness of $\varepsilon$ [13], and therefore an efficient implementation of implicit terms is vital.

[1] For the example of the Vlasov equation this choice can be expressed as follows. One might derive a separate model in the collisional limit by taking velocity moments of the particle distribution function and then providing a closure condition. This produces a set of equations for fluid quantities, dependent on space but not velocity space. Alternatively, one might resolve to solve the Vlasov equation directly, even though doing so entails solving a six-dimensional kinetic system, rather than three-dimensional fluid system.

Asymptotic preserving schemes were introduced in the 1990s for calculations of neutron transport [14, 15], and since schemes for kinetic systems relevant to Project NEPTUNE have been developed (see [12] for a review). Moreover, recent authors [16, 17] have developed high-order asymptotic preserving IMEX schemes. These schemes take a multiderivative approach, that is, unlike methods like Runge–Kutta or multistep, the methods also use higher time derivatives of the unknowns. While this makes the scheme more complicated, it also means the scheme is more local, incorporating more information from each timestep, and therefore has the potential to reduce the memory overhead on HPC systems [16].

### 2.1.2 Variable Stepsize, Variable Order

Variable Stepsize, Variable Order (VSVO) schemes are a family of schemes motivated by the idea of applying a "time filter" to the approximate solution as a means of increasing the order-of-accuracy of the original scheme. Time filters are themselves motivated by an *ad hoc* technique used in geophysical fluid dynamics to reduce oscillations in leapfrog time integration. In the Robert–Asselin filter [18, 19], the approximate solution produced by the leapfrog scheme, $u_i^*$, is "filtered" as a post-processing step to produce the final approximation

$$u_i = u_i^* + \frac{\nu}{2}\left(u_{i+1}^* - 2u_i + u_{i-1}\right), \tag{4}$$

where $\nu \sim 0.1$ is a free parameter.

Guzel & Layton [20] observed that, when using this filter as an on-the-fly corrector to a low order scheme, judicious choice of $\nu$ could increase the order-of-accuracy of the overall scheme. For example, first order implicit Euler,

$$u_{i+1}^* = u_i + \Delta t\, R\left(u_{i+1}^*, t\right), \tag{5}$$

becomes second-order accurate when filtered with

$$u_{i+1} = u_{i+1}^* - \frac{\nu}{2}\left(u_{i+1}^* - 2u_i + u_{i-1}\right), \tag{6}$$

with the specific value $\nu = 2/3$ (note also the change in sign of $\nu$ and shift in index relative to (4)). This particular method is in fact second-order backwards differentiation (BDF2), though in general the procedure does not produce BDF schemes [21].

Guzel & Layton [20] also generalized this approach for non-uniform timestep, while DeCaria [21] extended it to produce VSVO methods. For example, VSVO-12 adaptively switches between using the first order approximation $u_{i+1}$ and the second order approximation $u_{i+1}^*$ based on error estimates, and if neither meets a given tolerance adapts the timestep.

These methods are attractive because they offer a means to increase the order-of-accuracy of existing implementations with very little coding overhead. For example, given an implementation of implicit Euler (5), a second-order scheme could be obtained simply by implementing the time filter (6) and by storing one additional time slice $u_{i-1}$. However, the filter required is specific to the integration method used, so that each different method requires careful derivation of a different filter.

## 2.2 Elliptic systems

NEPTUNE-relevant systems will require the solution of elliptic equations in three dimensions and in complex geometries that exhibit strong anisotropy in the direction parallel to the magnetic field line. While multigrid's scalability makes it typically the favoured method for solving such problems on distributed systems, special care is needed when treating multiscale systems or strongly irregular meshes. Moreover, iterative methods (such as the smoothing step in multigrid) may require the underlying equations to be modified in order to deal with strong anisotropy.

In this section we discuss the solution of elliptic systems particularly in regard to two approaches: asymptotic preserving schemes that address the strong anisotropy, and nested solvers that address the complex geometries.

### 2.2.1 Asymptotic Preserving schemes

Asymptotic preserving schemes are also relevant in the spatial component of NEPTUNE-relevant problems where the strong magnetic field means that the system to be inverted is highly anisotropic.

A typical system like that studied by [22] may be written as

$$-\frac{1}{\varepsilon}\Delta_{\parallel}u^{\varepsilon} - \Delta_{\perp}u^{\varepsilon} = f \qquad \text{on } \Omega, \tag{7}$$

$$u^{\varepsilon} = 0 \qquad \text{on } \partial\Omega, \tag{8}$$

where $\Delta_{\parallel}$ and $\Delta_{\perp}$ respectively denote the parts of the Laplacian parallel and perpendicular to the strong magnetic field. The limiting system as $\varepsilon \to 0$ is

$$-\Delta_{\parallel}u^{0} = 0 \qquad \text{on } \Omega, \tag{9}$$

$$u^{0} = 0 \qquad \text{on } \partial\Omega, \tag{10}$$

which is degenerate, admitting an infinite number of solutions. Usual discretization methods are ill-conditioned for small $\varepsilon$, limiting anisotropies that can be studied. Instead asymptotic preserving discretizations must be derived; this is the subject of ongoing work [22, 23, 24, 25].

The two-field approach proposed in [22] eliminates the stiff terms by introducing the auxiliary field $q$, defined by $\varepsilon\Delta_{\parallel}q = \Delta_{\parallel}u$. With this, the strongly anisotropic system (7) may be split into two mildly anisotropic systems

$$\begin{aligned} -(\Delta_{\parallel} + \varepsilon_0\Delta_{\perp})u &= \varepsilon_0 f + (\varepsilon_0 - \varepsilon)\Delta_{\parallel}q \qquad \text{on } \Omega \\ u &= 0 \qquad \text{on } \partial\Omega \end{aligned} \tag{11}$$

$$\begin{aligned} -(\Delta_{\parallel} + \varepsilon_0\Delta_{\perp})q &= f + \Delta_{\perp}(u - \varepsilon_0 q) \qquad \text{on } \Omega \\ q &= 0 \qquad \text{on } \partial\Omega \end{aligned} \tag{12}$$

where $\varepsilon_0$ is a free parameter which should be chosen to satisfy $\varepsilon \ll \varepsilon_0 \ll 1$. This system is amenable to solution using an iterative method. Thus strongly anisotropic problems become tractable, but at the cost of solving for an additional field. Still, this extra cost is ameliorated by the fact that the operator to be inverted is the same in (11) and (12). Moreover, the system is

reasonably well-conditioned, with a condition number that scales as $1/(\varepsilon_0 h^2)$, improving on the $1/h^4$ scaling of earlier methods. The freedom to choose $\varepsilon_0$ also alleviates the problem of "locking", the phenomenon where the accuracy of a numerical approximation deteriorates as a parameter approaches a limiting value (here $\varepsilon \to 0$) [26]. Finally, the method does not place any constraint on the discretization of the spatial operators. This means it can be used in arbitrary magnetic fields, and also permits the method's use with closed magnetic field lines. This is in contrast to approaches based on field line integration [24, 25] which typically require analytic expressions for the magnetic field.

### 2.2.2 Nested solvers

Some mathematical descriptions of physical systems possess a natural hierarchy or structure that may be exploited by the numerical scheme or its implementation on HPC systems. For example, finite element methods contain matrix systems to be solved both within and between the basis elements. The system within an element is typically dense, while that coupling elements is typically sparse. Such a situation invites the use of different, nested numerical methods for the different systems. Moreover, modern HPC systems are typically hierarchical in their architecture, with multiple processors grouped together in nodes or NUMA regions. This suggests that the numerical schemes should be implemented respecting the structure of the HPC architecture. For example, the communication cost of a finite element method could be minimized by implementing the dense inner problem locally to a core (or, with a shared memory paradigm, locally to a NUMA region), while the sparser outer problem would be allowed to span cores (or NUMA regions).

Such an approach could also be applied problems where unstructured meshes lead to different spatial regions with different stiffness. One path to avoiding unfavourable mesh structures with multigrid would be to confine difficult mesh regions to a single inner problem and use a non-multigrid solver for that subdomain. While there is a developed literature on nested solvers (e.g. [27, 28, 29]), we are not aware of algorithms which subdivide the domain between solvers in this fashion.

# 3 Summary

This report describes recent advances in algorithms for solving hyperbolic and elliptic systems. Of particular interest are Implicit-Explicit (IMEX) schemes, which allow for the simultaneous advance of fast and slow timescales typically found in multiphysics systems, and asymptotic preserving schemes, which can efficiently treat the extreme anisotropy caused by the tokamak's strong magnetic field.

## Acknowledgement

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

## References

[1] S. Thorne. Priority Equations and Test Cases. Technical Report 2047353-TN-01, UKAEA Project Neptune, 2021.

[2] V. Alexandrov, A. Lebedev, E. Sahin, and S. Thorne. Linear systems of equations and preconditioners relating to the NEPTUNE Programme: A brief overview. Technical Report 2047353-TN-02, UKAEA Project Neptune, 2021.

[3] W. Arter, L. Anton, D. Samaddar, and R. Akers. ExCALIBUR Fusion Modelling System Science Plan. Technical Report CD/EXCALIBUR-FMS/0001, UKAEA, 2019. `https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/spf/ukaea-excalibur-fms-scienceplan.pdf`.

[4] H. Zhang, A. Sandu, and S. Blaise. Partitioned and implicit–explicit general linear methods for ordinary differential equations. *Journal of Scientific Computing*, 61(1):119–144, 2014.

[5] T. Hagstrom and R. Zhou. On the spectral deferred correction of splitting methods for initial value problems. *Communications in Applied Mathematics and Computational Science*, 1(1):169–205, 2007.

[6] S. Boscarino, J.-M. Qiu, and G. Russo. Implicit-explicit integral deferred correction methods for stiff problems. *SIAM Journal on Scientific Computing*, 40(2):A787–A816, 2018.

[7] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.

[8] A. Christlieb, B. Ong, and J.-M. Qiu. Integral deferred correction methods constructed with high order runge–kutta integrators. *Mathematics of Computation*, 79(270):761–783, 2010.

[9] K. Böhmer and H. J. Stetter. Defect correction methods. *Theory and applications*, 5, 1984.

[10] R. D. Skeel. A theoretical framework for proving accuracy results for deferred corrections. *SIAM Journal on Numerical Analysis*, 19(1):171–196, 1982.

[11] A. Christlieb, M. Morton, B. Ong, and J.-M. Qiu. Semi-implicit integral deferred correction constructed with additive runge–kutta methods. *Communications in Mathematical Sciences*, 9(3):879–902, 2011.

[12] S. Jin. Asymptotic preserving (AP) schemes for multiscale kinetic and hyperbolic equations: a review. *Rivista di Matematica della Università di Parma*, 3(January 2010):177–216, 2012.

[13] C. W. Gear. Numerical initial value problems in ordinary differential equations. *Prentice-Hall series in automatic computation*, 1971.

[14] E. W. Larsen, J. E. Morel, and W. F. Miller Jr. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes. *Journal of Computational Physics*, 69(2):283–324, 1987.

[15] E. W. Larsen and J. E. Morel. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes II. 1989.

[16] J. Schütz and D. C. Seal. An asymptotic preserving semi-implicit multiderivative solver. *Applied Numerical Mathematics*, 160:84–101, 2021.

[17] A. J. Dittmann. High-order multiderivative IMEX schemes. *Applied Numerical Mathematics*, 160:205–216, 2021.

[18] A. Robert. The integration of a spectral model of the atmosphere by the implicit method. In *Proc. WMO/IUGG Symposium on NWP, Tokyo, Japan Meteorological Agency*, volume 7, pages 19–24, 1969.

[19] R. Asselin. Frequency filter for time integrations. *Monthly Weather Review*, 100(6):487–490, 1972.

[20] A. Guzel and W. Layton. Time filters increase accuracy of the fully implicit method. *BIT Numerical Mathematics*, 58(2):301–315, jun 2018.

[21] V. DeCaria. Variable stepsize, variable order methods for partial differential equations. September 2019.

[22] F. Deluzet and J. Narski. A two field iterated asymptotic-preserving method for highly anisotropic elliptic equations. *Multiscale Modeling & Simulation*, 17(1):434–459, 2019.

[23] L. Chacón, D. del Castillo-Negrete, and C. D. Hauck. An asymptotic-preserving semi-lagrangian algorithm for the time-dependent anisotropic heat transport equation. *Journal of Computational Physics*, 272:719–746, 2014.

[24] M. Tang and Y. Wang. An asymptotic preserving method for strongly anisotropic diffusion equations based on field line integration. *Journal of Computational Physics*, 330:735–748, 2017.

[25] Y. Wang, W. Ying, and M. Tang. Uniformly convergent scheme for strongly anisotropic diffusion equations with closed field lines. *SIAM Journal on Scientific Computing*, 40(5):B1253–B1276, 2018.

[26] I. Babuška and M. Suri. On locking and robustness in the finite element method. *SIAM Journal on Numerical Analysis*, 29(5):1261–1293, 1992.

[27] E. Polizzi and A. H. Sameh. A parallel hybrid banded system solver: the SPIKE algorithm. *Parallel computing*, 32(2):177–194, 2006.

[28] E. Laszlo, M. Giles, and J. Appleyard. Manycore algorithms for batch scalar and block tridiagonal solvers. *ACM Transactions on Mathematical Software (TOMS)*, 42(4):1–36, 2016.

[29] J. T. Parker, P. A. Hill, D. Dickinson, and B. D. Dudson. Parallel tridiagonal matrix inversion with a hybrid multigrid-thomas algorithm method. *Journal of Computational and Applied Mathematics*, 399:113706, 2022.