Multiscale-Optimized Plasma Turbulence Simulation on Petascale Architectures

J. Candy[a], I. Sfiligoi[c], E. Belli[a], K. Hallatschek[b], C. Holland[c], N. Howard[d], E. D'Azevedo[e]

[a]*General Atomics, San Diego, CA*
[b]*Max-Planck-Institute for Plasma Physics, Garching, Germany*
[c]*University of California at San Diego, San Diego, CA*
[d]*MIT Plasma Science and Fusion Center, Cambridge, MA*
[e]*Oak Ridge National Laboratory, Oak Ridge, TN*

Abstract

We describe the mathematical formulation, outline the numerical discretization, and present performance analysis results for the CGYRO plasma turbulence code. The performance data was collected on 5 current leadership systems (2 KNL-based, 2 hybrid CPU-GPU and 1 Skylake-based). CGYRO is a relatively new gyrokinetic turbulence code, based on the well-known GYRO code, but redesigned from the ground up to operate efficiently on multicore and GPU-accelerated systems. The gyrokinetic equations specify a 5-dimensional distribution function for each species, with species coupled through both the Maxwell equations and collision operator. For the cross-machine performance analysis, we report and compare timings for 8 separate computational kernels. This kernel-based breakdown illustrates the strengths and weaknesses of the floating-point and communication architectures of the respective systems. We conclude with a preview of new multiscale turbulence results that are shown to accurately recover experimentally-observed electron turbulence levels in an ITER-baseline plasma regime that cannot be described using traditional long-wavelength simulation.

1. Fusion plasma as a low-carbon energy source

The US and global economies increasingly depend on reliable sources of energy. In coming decades, these sources must become increasingly low-carbon to mitigate the risks of climate change. Thus, the challenge to harness the virtually inexhaustible potential of fusion energy is being pursued in a coordinated worldwide effort. In parallel with a vibrant global research program (based primarily on the toroidal magnetic confinement, or tokamak, concept), numerical simulation serves as a powerful tool for accelerating progress. Simulations are used to validate basic theory, plan experiments, interpret results on present devices, and ultimately to design future devices. While the *long-term* goal of fusion simulation is to provide the scientific basis for a demonstration reactor, a *near-term* goal is to refine our understanding of physics issues associated with burning plasmas. This simulation capability relies on high-performance computing, enabling researchers to obtain key insights from fundamental physical models.

2. Theoretical framework for plasma turbulence

2.1. The Fokker-Planck model

Magnetically-confined plasmas obey the Fokker-Planck kinetic equation, including heat, particle and momentum sources, and constrained by nonlinear interspecies collisions [1, 2]. Long-wavelength (equilibrium) plasma dynamics are typically described using low-order fluid (magnetohydrodynamic, or MHD) moments of this kinetic equation. The theory shows that, in equilibrium, the fluid pressure is balanced by that of the confining magnetic field resulting in a nested set of *flux surfaces*: closed toroidal surfaces of constant magnetic flux. The slow transport of particles and energy across these surfaces is not captured by the MHD equations and must be described by kinetic theory that retains, via sophisticated multiple-scale analysis [2], the parallel and gyroaveraged drift motion. This analysis results in two well-known but separate equations: the *neoclassical equation* [3, 4] for the $\mathbf{k}_\perp = 0$ correction to the distribution, and the *gyrokinetic equation* [5] for the $\mathbf{k}_\perp > 0$ correction (small-scale fluctuations). The two theories are complementary, as shown by Sugama in the 1990s [2]. Because of this complementary condition, the neoclassical and gyrokinetic fluxes are summed to provide the total cross-surface transport flux. This flux is an input to self-consistent transport-timescale solvers like TGYRO [6, 7] that can evolve the plasma profiles by balancing the computed transport fluxes with input heating sources (ion beams, radio-frequency waves, thermonuclear alpha particles).

2.2. Gyrokinetic waves, instabilities and turbulence

The gyrokinetic (GK) equations describe stable and unstable kinetic plasma waves – typically referred to as *drift waves* [8], or *microinstabilities*. These waves grow and nonlinearly saturate, typically developing into a quasi-steady turbulent state that generates plasma transport and thus sets a limit on the energy confinement. A detailed understanding of this turbulence is required for the construction of validated predictive models [9, 10] that can be used to understand the performance of current-day experiments, and to further predict the performance of future experiments such as the International Thermonuclear Experimental Reactor (ITER) [11, 12]. The ability to accurately

simulate *electron energy transport* is particularly important for future fusion reactors because, in contrast to existing tokamaks with significant ion heating (via neutral beam injection), future burning plasmas will be dominantly heated by fusion products which preferentially heat electrons via collisional energy transfer.

2.3. The challenge of multiscale electron-ion coupling

Drift waves are observed to cover a broad range of spatiotemporal scales. For instance, ion-temperature-gradient (ITG) and electron-temperature-gradient (ETG) modes occur at ion and electron gyroradius scales, respectively, which differ by almost two orders of magnitude. Extensive validation studies using GK models over the last decade have shown the ability to accurately and routinely predict ion-scale energy fluxes in a variety of plasmas obtained in different experimental devices [13, 14, 15, 16, 17, 18, 19, 20]. These simulations typically resolve wavenumbers in the range $0 \leq k_y\rho_s < 1$ where ITG turbulence is (usually) the dominant source of ion transport. Here, k_y is the toroidal (or, binormal) wavenumber and ρ_s is the ion-sound gyroradius. In addition to predicting ion energy fluxes consistent with independent power balance analyses, these simulations have simultaneously predicted turbulent long-wavelength fluctuation spectra consistent with measurements [21, 16], further increasing confidence that the fundamental dynamics at these wavenumbers is being accurately captured.

However, simulation of multiscale turbulence – that is, turbulence spanning both ion and electron scales *seamlessly* – is far more challenging and presently only feasible with leadership-class computational resources. Importantly, multiscale simulation requires an *arbitrary-wavelength formulation* of the equations and algorithms, in contrast to long-wavelength approximations nearly ubiquitous among GK codes. The earliest exploratory multiscale simulations were carried out (with reduced ion-to-electron mass ratio) a decade ago using a Cray X1E supercomputer [22]. Significantly more recent and expensive multiscale simulations that retain the correct mass ratio [23, 24, 25, 26] demonstrate that self-consistent inclusion of ETG modes can lead to significant levels of electron energy transport on electron scales, and can increase ion-scale turbulence and associated long-wavelength ion and electron energy transport. For other values of input parameters (consistent with experimental measurements and their uncertainties), a suppression of ion-scale turbulence driven by electron-scale turbulence has also been observed [25]. To date, multiscale turbulence simulations with physical mass ratio have been reported by only one European (GENE) and one US (GYRO) research group. Thus, while turbulence in inner (or core) plasma regions has been extensively studied over the last decade, there are only a handful of reported multiscale simulations.

2.4. CGYRO: a new multiscale-optimized solver

Over the last decade the fusion community has focused its modeling efforts primarily on the core region. A popular kinetic code used for this purpose was GYRO [27, 28, 29, 30]. Thousands of nonlinear simulations with GYRO have informed the fusion community's understanding of core plasma turbulence [31, 32, 33, 24] and provided a *transport database* for the calibration of reduced transport models such as TGLF [9]. GYRO was the first global electromagnetic solver, and pioneered the development of numerical algorithms for the GK equations with kinetic electrons. It is formulated in real space and like all global solvers requires *ad hoc* absorbing-layer boundary conditions when simulating cases with profile variation. This approach is suitable for core turbulence simulations, which cover a large radial region and are dominated by low wavenumbers. More recently, as the understanding of core transport has become increasingly complete, the cutting edge of research moved radially toward the pedestal region, where plasmas are characterized by larger collisionality and steeper pressure gradients that greatly modify the turbulent phenomena at play. This motivated the development, from scratch, of the CGYRO code [34, 35, 36, 37] to complement GYRO. CGYRO, the focus of this report, is an Eulerian GK solver specifically designed and optimized for collisional, electromagnetic, multiscale simulation. A critical algorithmic aspect of CGYRO is the radially spectral formulation used to reduce the complicated integral gyroaveraging kernel into a multiplication in wavenumber space, but retaining the ability to treat profile variation important for edge plasmas. A new coordinate system that is more suitable for the highly collisional and shaped edge regime was adopted from the NEO code [38, 39], which is the community standard for calculation of collisional transport in toroidal geometry.

3. Computational Approach

CGYRO is compliant with the Fortran 2008 standard and was designed to be suitable for next-generation computational systems that require high levels of parallel concurrency. The implementation combines 15 years of algorithmic lessons learned from GYRO, together with an array distribution scheme and loop structure that targets modern multicore and accelerated (GPU) architectures. The key computational kernels in CGYRO have been optimized independently for multicore and GPU-based systems, and the code benchmarked against GYRO in the limit of weak collisions and rotation. For strong collisions and rapid plasma rotation, however, CGYRO is more realistic as it implements the complete *Sugama electromagnetic gyrokinetic theory* [2]. The GACODE build system [40] is used to ensure portability, with the code fully operational at ALCF Mira/Theta, OLCF Titan, CSCS Piz Daint, TACC Stampede2, NERSC Cori, and elsewhere. In this section we describe the underlying equations solved, together with the structure of the various kernels.

3.1. The gyrokinetic model

The nonlinear, electromagnetic GK equations specify 5D particle distributions for electron and multiple ion species:

$$\widetilde{H}_a(k_x, k_y, \theta; \xi, \mathrm{v}) \,, \tag{1}$$

where the subscript a is the species index, and the tilde indicates a Fourier space quantity. The *spatial coordinates* are

$$k_x \longrightarrow \text{radial wavenumbers} \tag{2}$$
$$k_y \longrightarrow \text{binormal wavenumbers} \tag{3}$$
$$\theta \longrightarrow \text{field-line coordinate} \tag{4}$$

where $k_\perp^2 = k_x^2 + k_y^2$, and the *velocity-space* coordinates are

$$\xi = \mathrm{v}_\parallel/\mathrm{v} \longrightarrow \text{cosine of the pitch angle} \in [-1, 1] \tag{5}$$
$$\mathrm{v} \longrightarrow \text{speed} \in [0, \infty] \; . \tag{6}$$

Because of the use of twisted fieldline coordinates, the radial wavenumbers k_x depend on θ and k_y [34]. For this reason, it is convenient to define a primitive radial wavenumber k_x^0 (the value of k_x at $\theta = 0$) that can be directly quantized (in CGYRO, we write $k_x^0 = 2\pi p/L$ where p is an integer, and L is the radial domain size). A schematic illustration of the 5D (plus species) mesh is given in Fig. 1 for resolution typical of a multiscale simulation. This figure also shows a sub-mesh typical of standard low-$k_\perp$ ion-scale simulation. The spectral representation in terms of (k_x, k_y) is key to the arbitrary wavelength formulation and diagonalizes the gyroradius dynamics. Despite the use of a spectral representation, slow (global) variation of the plasma profiles are (optionally) retained using a new *wavenumber advection* algorithm [36]. The GK equations are written in terms of an electromagnetic *field potential* $\widetilde{\Psi}_a$, defined as

$$\widetilde{\Psi}_a = J_0(k_\perp\rho_a)\left(\delta\widetilde{\phi} - \frac{\mathrm{v}_\parallel}{c}\delta\widetilde{A}_\parallel\right) + \frac{m_a\mathrm{v}_\perp^2}{z_a e B}\frac{J_1(k_\perp\rho_a)}{k_\perp\rho_a}\delta\widetilde{B}_\parallel \; , \tag{7}$$

where m_a, z_a and ρ_a are the species mass, charge and gyroradius. Above, $(\delta\widetilde{\phi}, \delta\widetilde{A}_\parallel, \delta\widetilde{B}_\parallel)$ are the electrostatic, transverse electromagnetic, and compressional electromagnetic potentials respectively, computed via coupling with the Maxwell equations. The Bessel functions J_0 and J_1 in Eq. (7) arise from gyroaveraging. This simple, compact representation of the field potential $\widetilde{\Psi}_a$ (and the Maxwell equations that we will write shortly) is only possible using a spectral wavenumber expansion. In terms of $\widetilde{\Psi}_a$, the GK equation for species a is written symbolically as

$$\frac{\partial\widetilde{h}_a}{\partial\tau} + A(\widetilde{H}_a, \widetilde{\Psi}_a) + B(\widetilde{H}_a, \widetilde{\Psi}_a) = 0 \; , \tag{8}$$

with $\tau \doteq (c_s/a)t$ the normalized time, a the midplane minor radius of the last closed flux surface, $c_s = \sqrt{T_e/m_D}$ the deuteron sound speed, T_e the electron temperature and m_D the deuteron mass. $A(\widetilde{H}_a, \widetilde{\Psi}_a)$ represents the collisionless terms and $B(\widetilde{H}_a, \widetilde{\Psi}_a)$ represents the collisional terms. The collisionless term $A(\widetilde{H}_a, \widetilde{\Psi}_a)$ includes the streaming motion along the magnetic field line, the drifts, the gradient drive due to equilibrium-scale density and temperature inhomegeneities, and the nonlinearity. It is described in more detail in Section 3.4. The collisional term $B(\widetilde{H}_a, \widetilde{\Psi}_a)$ includes the mixing in velocity space due to pitch angle scattering and energy diffusion due to binary collisions and also includes particle trapping. It is described in more detail in Section 3.5. Note that in Eq. (8), the function $\widetilde{h}_a$ is evolved rather than $\widetilde{H}_a$. Physically, the function $\widetilde{H}_a$ is the *nonadiabatic distribution*, which is the theoretical quantity of interest, whereas $\widetilde{h}_a$ is a *modified distribution* more suitable for numerical time-integration. They are related through the field potential by

$$\widetilde{h}_a = \widetilde{H}_a - \frac{z_a T_e}{T_a}\widetilde{\Psi}_a \; . \tag{9}$$

The time-independent gyrokinetic Maxwell equations, which relate the field potential to velocity-space integrals of $\widetilde{H}_a$, are

$$\left(k_\perp^2\lambda_D^2 n_e + \sum_a \frac{z_a^2 T_e}{T_a} n_a\right)\delta\widetilde{\phi} = \sum_a z_a e \int d^3\mathrm{v} f_{0a} J_0(k_\perp\rho_a)\widetilde{H}_a \; , \tag{10}$$

$$\frac{2n_e}{\beta_{e,\mathrm{unit}}} k_\perp^2\rho_s^2\delta\widetilde{A}_\parallel = \sum_a z_a e \int d^3\mathrm{v}\frac{\mathrm{v}_\parallel}{c_s} f_{0a} J_0(k_\perp\rho_a)\widetilde{H}_a \; , \tag{11}$$

$$-\frac{2n_e}{\beta_{e,\mathrm{unit}}}\frac{B}{B_{\mathrm{unit}}}\delta\widetilde{B}_\parallel = \sum_a \int d^3\mathrm{v}\frac{m_a\mathrm{v}_\perp^2}{T_e} f_{0a}\frac{J_1(k_\perp\rho_a)}{k_\perp\rho_a}\widetilde{H}_a \; . \tag{12}$$

Here $\lambda_D = \sqrt{T_e/(4\pi n_e e^2)}$ is the Debye length and $\beta_{e,\mathrm{unit}} = 8\pi n_e T_e/B_{\mathrm{unit}}^2$ is the effective electron beta, where B_{unit} is the effective magnetic field [30]. For a detailed survey of electromagnetic drift-wave instabilities we refer the reader to Ref. [41].

3.2. Suitability of Eulerian methods

The CGYRO velocity-space coordinates are efficient for plasmas with finite collision rate, and were patterned after the coordinates used in the successful neoclassical code, NEO [38]. The numerical discretization is *spectral* in (k_x, k_y), *pseudospectral* in (ξ, v) and uses a unique 5th-order *conservative upwind scheme* in θ. The upwind scheme was constructed to ensure high-accuracy electromagnetic calculation even for high plasma β and vanishingly small perpendicular wavenumber, $k_\perp \to 0$. It is perhaps not well-understood that only Eulerian algorithms, in contrast to Lagrangian (PIC) methods, can accurately treat the high-β, low-$k_\perp$ regime without electron-fluid approximations or numerical inaccuracy constraints [42]. To our knowledge, only Eulerian solvers have treated arbitrary fluctuation wavenumbers electromagnetically (without resorting to Padé, 4-point or related approximations). But this generality comes at a cost: multiscale Eulerian simulations that treat ion-scale and electron-scale turbulence simultaneously require an *extremely fine spatial mesh* – and therefore specialized numerical schemes – to prevent severe bottlenecks related to gyroaveraging and solution of the Maxwell equations. These bottlenecks do not arise for traditional ion-scale GK simulations with $k_\perp\rho_i < 1.0$.

3.3. Time advance

An operator splitting scheme is used to separate the collisionless term $A(\widetilde{H}_a, \widetilde{\Psi}_a)$ (streaming, drifts, gradient drive, and nonlinearity) from the collisional term $B(\widetilde{H}_a, \widetilde{\Psi}_a)$ (collisional diffusion in velocity space). This allows the nonlinear, collisionless dynamics to be treated with an *explicit* time advance, and the collisional dynamics to be treated with an *implicit* time advance.

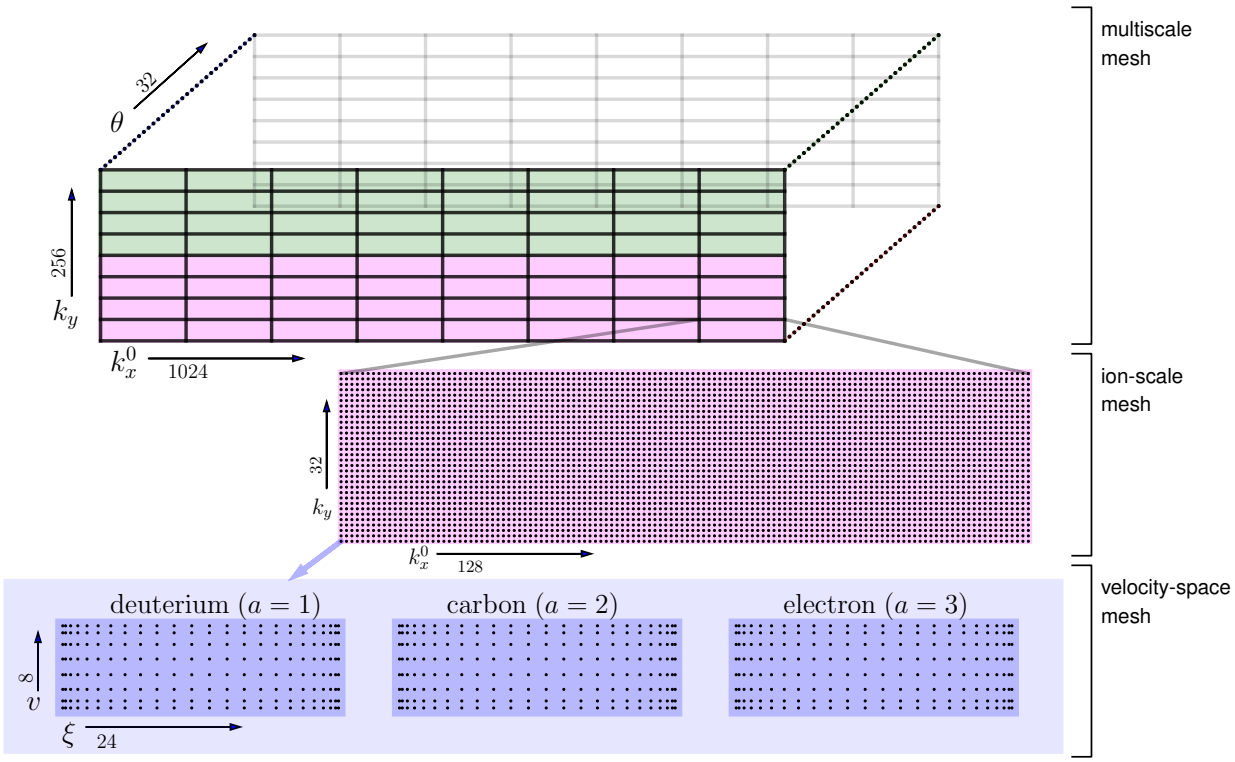

Figure 1: Illustration of the CGYRO 5D mesh using a resolution that is rougly typical of a multiscale simulation. The relation of the multiscale wavenumber domain (1024×256) to an ion-scale domain (128×32) is also illustrated.

3.4. Collisionless step

This step operates primarily on the spatial dimensions and is distributed in the velocity dimensions. The collisionless step requires solution of the equation

$$\frac{\partial \widetilde{h}_a}{\partial \tau} + A(\widetilde{H}_a, \widetilde{\Psi}_a) = 0 \,, \tag{13}$$

and uses an explicit time advance (RK4). We write the collisionless term symbolically as:

$$A(\widetilde{H}_a, \widetilde{\Psi}_a) = -i\left(\Omega_{\text{parallel}} + \Omega_{\text{drift}}\right)\widetilde{H}_a - i\Omega_* \widetilde{\Psi}_a - \frac{c}{B}\frac{a}{c_s}\sum_{\mathbf{k}'_\perp + \mathbf{k}''_\perp = \mathbf{k}_\perp} (\mathbf{b}\cdot \mathbf{k}'_\perp \times \mathbf{k}''_\perp)\widetilde{\Psi}_a(\mathbf{k}'_\perp)\widetilde{h}_a(\mathbf{k}''_\perp) \,. \tag{14}$$

The linear terms in $A(\widetilde{H}_a, \widetilde{\Psi}_a)$ include the parallel streaming along the field line,

$$-i\Omega_{\text{parallel}} = \frac{\mathrm{v}_\parallel}{w_s}\frac{\partial}{\partial \theta} \,, \tag{15}$$

the drift motion perpendicular to the field line,

$$-i\Omega_{\text{drift}} = i\mathbf{k}_\perp \rho_a \cdot \frac{\mathrm{v}^2}{2\mathrm{v}_{ta}}\frac{a}{c_s}\left[\left(1+\xi^2\right)\frac{\mathbf{b}\times\nabla B}{B} + \xi^2 \frac{8\pi}{B^2}\frac{dp}{dr}\mathbf{b}\times\nabla r\right] \,, \tag{16}$$

and *instability drive* from equilibrium density and temperature gradients

$$-i\Omega_* = -ik_\theta \rho_s a\left[\frac{d\ln n_a}{dr} + \frac{d\ln T_a}{dr}\left(\frac{\mathrm{v}^2}{2\mathrm{v}_{ta}^2} - \frac{3}{2}\right)\right] \,. \tag{17}$$

Here $w_s = c_s(\mathcal{J}_\psi B)/a$ is an effective velocity (with $\mathcal{J}_\psi$ is the Jacobian determinant), $\mathbf{b} = \mathbf{B}/B$, p is the total pressure and $\mathrm{v}_{ta} = \sqrt{T_a/m_a}$ is the thermal speed. We further define the gyroradius $\rho_a = \mathrm{v}_{ta}/\Omega_{ca}$, where Ω_{ca} is the gyrofrequency of species a, and the effective ion-sound gyroradius

$$\rho_s = \frac{c_s}{eB_{\text{unit}}/(m_D c)} \,.$$

The wavenumber k_θ is related to k_y (see Ref. [34]). The linear terms in $A(\widetilde{H}_a, \widetilde{\Psi}_a)$, namely Ω_{parallel}, Ω_{drift}, and Ω_*, define the *streaming kernel*, hereafter referred to as `str`. The last term in Eq. (14) is a type of convolution (a Poisson bracket in real space). This defines the *nonlinear kernel* and is hereafter referred to as `nl`. Finally, note that explicit coupling with the Maxwell equations is also required to advance $\widetilde{\Psi}_a$ in time. This operation defines the *field solve* kernel, hereafter referred to as `field`. In this report we consider only the limit of zero plasma rotation. The case of sonic rotation is described in Ref. [37].

3.5. Collisional step

The collisional step acts primarily on the velocity dimensions and is distributed in the spatial dimensions:

$$\frac{\partial \widetilde{h}_a}{\partial \tau} + B(\widetilde{H}_a, \widetilde{\Psi}_a) = 0 \,, \tag{18}$$

where

$$B(\widetilde{H}_a, \widetilde{\Psi}_a) = -i\Omega_\xi \widetilde{H}_a - \frac{a}{c_s}\sum_b C^L_{ab}(\widetilde{H}_a, \widetilde{H}_b)\ .$$

Here $-i\Omega_\xi$ is a linear term representing the trapped particle dynamics,

$$-i\Omega_\xi = -\frac{\mathrm{v}}{2w_s}\left(1-\xi^2\right)\frac{\partial \ln B}{\partial\theta}\frac{\partial}{\partial\xi}\ , \tag{19}$$

and C^L_{ab} is the linearized gyrophase-averaged collision operator. CGYRO implements an advanced collision operator, beyond that used in standard gyrokinetics. This takes the form

$$\begin{aligned} C^L_{ab}(\widetilde{H}_a, \widetilde{H}_b) =& \frac{\nu^D_{ab}}{2}\frac{\partial}{\partial\xi}\left(1-\xi^2\right)\frac{\partial\widetilde{H}_a}{\partial\xi} \\ &+ \frac{1}{\mathrm{v}^2}\frac{\partial}{\partial\mathrm{v}}\left[\frac{\nu^{\parallel}_{ab}}{2}\left(\mathrm{v}^4\frac{\partial\widetilde{H}_a}{\partial\mathrm{v}} + \frac{m_a}{T_b}\mathrm{v}^5\widetilde{H}_a\right)\right] \\ &- \widetilde{H}_a k_\perp^2 \mathcal{F}(\mathrm{v},\xi) + R_{\mathrm{m}}(\widetilde{H}_b) + R_{\mathrm{e}}(\widetilde{H}_b)\ , \end{aligned} \tag{20}$$

with terms describing pitch-angle diffusion, energy diffusion, non-diffusive finite Larmor radius corrections, momentum conservation, and energy conservation, respectively. Here, $\nu^D_{ab}(\mathrm{v})$ is the pitch-angle diffusion (deflection) rate and $\nu^{\parallel}_{ab}(\mathrm{v})$ is the parallel velocity diffusion rate. A Legendre pseudospectral discretization in ξ is combined with a Hermite-like pseudospectral discretization in v. Using a weak form of the discrete collision operator (matrix), we construct a manifestly self-adjoint form in terms of the Gaussian weights and the pseudospectral derivative matrices. We have tested the scheme for collision frequencies 10^4 times greater than the highest expected values in a tokamak with no sign of instability or unphysical behavior. Rewriting the Eq. (18) in terms of $\widetilde{H}_a$, we find

$$\frac{\partial\widetilde{H}_a}{\partial\tau} - \frac{z_a T_e}{T_a}\frac{\partial\widetilde{\Psi}_a}{\partial\tau} - i\Omega_\xi\widetilde{H}_a = \frac{a}{c_s}\sum_b C^L_{ab}(\widetilde{H}_a, \widetilde{H}_b)\ .$$

When retaining the field potential $\widetilde{\Psi}_a$, it is (to our knowledge) necessary to use an implicit time-advance if stability is desired without severe accuracy loss. Using a generalization of the Crank-Nicolson method, we advance this equation with a single matrix-vector multiply (matrix rank $N_\xi N_v N_a$). Although this is a large matrix, the problem can be easily distributed over the entire spatial mesh and is thus straightforward to parallelize. For the scaling results presented in the present work, we use a partially simplified operator to reduce memory usage. Note however that the exact collision operator exhibits *better* scalability (via higher ratio of computation to communication) than the simplified operator. The simplified operator is nevertheless useful for systems with limited memory or for cases where collisions are relatively weak. This *collision kernel* is hereafter referred to as `coll`.

3.6. FFTW/cuFFT-based evaluation of the nonlinearity

The treatment of the quadratic nonlinearity, through numerical evaluation of the convolution given in Eq. (14), is done in a standard way using a 2D Fast-Fourier transform (FFT) with dealiasing [43]. The convolution can be evaluated by direct summation (and pruning unresolved wavenumbers) but to do so would be prohibitively slow for typical nonlinear simulations. Alternatively, one performs a forward transform, multiplies the functions in real-space, followed by the inverse transform. Uncorrected, this procedure gives rise to *aliasing*. To prevent aliasing we first zero-pad the spectral representation by a factor of 3/2, take the forward transform to a finer real-space mesh, multiply, take the inverse transform, and retain only the original wavenumbers. The dealiased convolution conserves important flow invariants and eliminates a class of nonlinear instabilities from the numerical solution. To perform the forward and inverse FFTs, we use `FFTW` [44] by default with options for cuFFT (GPU) on Titan and Intel MKL on supported platforms. More specifically, in CGYRO we perform a series of four 2D complex-to-real (`c2r`) transforms

$$(ik_x)\widetilde{\Psi}_a \underset{\texttt{c2r}}{\longrightarrow} \frac{\partial\Psi_a}{\partial x}\ , \quad (ik_x)\widetilde{h}_a \underset{\texttt{c2r}}{\longrightarrow} \frac{\partial h_a}{\partial x}\ , \tag{21}$$

$$(ik_y)\widetilde{\Psi}_a \underset{\texttt{c2r}}{\longrightarrow} \frac{\partial\Psi_a}{\partial y}\ , \quad (ik_y)\widetilde{h}_a \underset{\texttt{c2r}}{\longrightarrow} \frac{\partial h_a}{\partial y}\ , \tag{22}$$

where x and y are effective real-space meshpoints, such that all arrays are extended and zero-padded by a factor of 3/2 (quantities without tildes are in real space). The real-space products are then taken, followed by the inverse transform of the entire nonlinearity via a single 2D real-to-complex (`r2c`) transform

$$-\frac{\partial\Psi_a}{\partial x}\frac{\partial h_a}{\partial y} + \frac{\partial h_a}{\partial x}\frac{\partial\Psi_a}{\partial y} \underset{\texttt{r2c}}{\longrightarrow} (\mathbf{b}\cdot\mathbf{k}'_\perp\times\mathbf{k}''_\perp)\,\widetilde{\Psi}_a(\mathbf{k}'_\perp)\widetilde{h}_a(\mathbf{k}''_\perp)\ . \tag{23}$$

3.7. Array layouts and communication

From the computational point of view, there are three array layouts. Two are associated with the linear terms, and the third with the nonlinear kernel. Internally, we define *lumped* variables for convenience. That is, we label the lumped configuration pair (k^0_x, θ) as a single array with dimension $\texttt{nc} = N_x \times N_\theta$, and the lumped velocity triplet (ξ, v, a) as a single array with dimension $\texttt{nv} = N_\xi \times N_\mathrm{v} \times N_a$. In the binormal direction, N_y values of k_y are simulated, with the h_a for different values of k_y *independent* in the absence of nonlinear coupling. This has important implications for optimization of linear simulations.

First, there is a **collisionless layout** for the linear terms in $A(\widetilde{H}_a, \widetilde{\Psi}_a)$ with all of configuration space (`nc` gridpoints) on an MPI task, but distributed in velocity space (`nv` gridpoints) on communicator 1 and in k_y on communicator 2 (with a single k_y per task):

$$\texttt{h(ic, iv\_loc)} \longrightarrow \underbrace{k^0_x, \theta}_{\texttt{ic}}\ , [k_y]_2, \underbrace{[\xi, \mathrm{v}, a]_1}_{\texttt{iv\_loc}}\ . \tag{24}$$

Note that there is no distributed index associated with k_y because, as noted above, there is exactly *one* value of k_y for a given MPI task. Next, there is a **collisional layout** for $B(H_a, \Psi_a)$ with all of velocity space on an MPI task, but distributed in configuration space:

$$\texttt{h(ic\_loc, iv)} \longrightarrow \underbrace{[k^0_x, \theta]_1}_{\texttt{ic\_loc}}, [k_y]_2, \underbrace{\xi, \mathrm{v}, a}_{\texttt{iv}}\ . \tag{25}$$

Finally, there is a **nonlinear layout**

$$\mathtt{h(ir,in,j\_loc)} \longrightarrow \underbrace{k_x^0}_{\mathtt{ir}} \, , \underbrace{k_y}_{\mathtt{in}} \, , \underbrace{[\theta,[\xi,\mathrm{v},a]_1]_2}_{\mathtt{j\_loc}} \, . \qquad (26)$$

Two dominant types of communication are required. To switch from the collisionless layout to the collisional layout and back to perform the collisional step, we require a *collision communication*, or `coll_comm`. To treat the nonlinearity in $A(\widetilde{H}_a, \widetilde{\Psi}_a)$, the linear process grid is multiplied by N_y and all toroidal modes are collected on a single core using the *nonlinear communication*, or `nl_comm`. These previous two `comm` operations are based on `MPI_ALLTOALL`, but only across a *single* (not both) MPI subcommunicators. A relatively inexpensive *field communication*, or `field_comm`, based on `MPI_ALLREDUCE`, is also required for solution of the gyrokinetic Maxwell equations, involving the RHS velocity-space integration of $\widetilde{H}_a$. Finally, there is a communication associated with the conservative upwind scheme, which we denote as `str_comm`.

In total, we have defined eight *computational kernels*, summarized for convenience in Table 1. Although CGYRO is capable of significant MPI parallelism (with n_{MPI} a multiple of N_y), the ability to take advantage of the multicore architecture of the Xeon Phi and other hardware through shared memory parallelism allows for the reduction of required communication relative to calculation. The on-node parallelization scheme for the most demanding kernel employs cache-aligned data arrays, OpenMP parallelized loops, and thread-safe FFT libraries. The hybrid strategy follows a standard pattern in the code. For example, in the collisionless layout the parallel code blocks take the generic form

```
!$omp parallel do private(jv,ic)
     do iv=nv1,nv2
        jv = iv-nv1+1
        do ic=1,nc
           f(ic,jv) = ...
        enddo
     enddo
```

The loop over `iv` has been distributed with MPI, so that the remaining work over `jv` (which is the local subset of `iv` indices) and the entire `ic` lumped index can be distributed with OpenMP. Directly analogous constructions are used for the other layouts.

4. Cross-platform Performance Analyses

CGYRO performance testing was carried out on five leadership systems:

1. NERSC Cori (KNL)
2. OLCF Titan
3. TACC Stampede2 (KNL)
4. TACC Stampede2 (Skylake)
5. CSCS Piz Daint

An overview of the key features of each architecture is given in Tables 2 (CPU/GPU systems) and 3 (CPU systems). To make inter-machine comparisons, one must have a meaningful *equal performance metric*. Because it can be misleading to compare multicore CPU systems (Cori, Stampede2, Skylake) to GPU-based system (Titan, Piz Daint) using a thread-to-thread comparison, we include comparisons using vendor peak-performance claims. This means, for example, that

two Titan nodes = one Cori KNL node = 3.0 peak TFLOP

Table 1: Summary of data properties of kernels. Here `str` refers to parallel streaming, `nl` refers to the nonlinear bracket (convolution), `field` refers to the solution of the three Maxwell equations, and `coll` refers to the implicit collision step. In each case, the communication cost associated with each kernel is denoted by the `comm` suffix.

Kernel	Data dependence	Dominant operation
`str`	$k_x^0, \theta, [k_y]_2, [\xi, \mathrm{v}, a]_1$	loop
`field`	Same as `str`	loop
`coll`	$[k_x^0, \theta]_1, [k_y]_2, \xi, \mathrm{v}, a$	mat-vec multiply
`nl`	$k_x^0, k_y, [\theta, [\xi, \mathrm{v}, a]_1]_2$	FFT
`str_comm`	$k_x^0, \theta, [k_y]_2, \underline{[\xi, \mathrm{v}, a]_1}$	MPI_ALLREDUCE
`field_comm`	Same as `str_comm`	MPI_ALLREDUCE
`coll_comm`	$k_x^0, \theta, [k_y]_2, [\xi, \mathrm{v}, a]_1$ $\longleftrightarrow [k_x^0, \theta]_1, [k_y]_2, \xi, \mathrm{v}, a$	MPI_ALLTOALL
`nl_comm`	$k_x^0, \theta, [k_y]_2, [\xi, \mathrm{v}, a]_1$ $\longleftrightarrow k_x^0, k_y, [\theta, [\xi, \mathrm{v}, a]_1]_2$	MPI_ALLTOALL

Table 2: Architecture overview of hybrid CPU/GPU systems including theoretical peak. TFLOP/node will be used as a normalizing factor in the performance analysis.

	`Titan`	`Piz Daint`
Architecture	CPU/GPU	CPU/GPU
CPU Model	Opteron 6274	Xeon ES-2690 v3
GPU Model	Tesla K20X 6GB	Tesla P100 16GB
Threads/node	16/2688	12/3584
TFLOP/node	1.5 (0.2+1.3)	4.5 (0.5+4.0)
Nodes	18688	5320
Interconnect	Cray Gemini	Cray Aries
Net. Topology	3D Torus	Dragonfly
Compiler	PGI Fort 17	PGI Fort 17
FFT library	cuFFT	cuFFT
MPI	Cray MPICH v7.6	Cray MPICH v7.6

4.1. Strong-scaling performance and total wallclock time

For system performance comparisons we collected timing data for a test case that is broadly representative of a *small multiscale simulation*, hereafter to by the abbreviation `nl03` . This case is available from the CGYRO command line via

```
$ cgyro -g nl03  .
```

The resolutions in each dimension are

$$(N_x, N_y, N_\theta, N_\xi, N_\mathrm{v}, N_a) = (512, 128, 32, 24, 8, 3) \, .$$

Note that in a subsequent section we will briefly examine performance on a larger case. Although `nl03` uses significantly higher radial (k_x^0) and binormal (k_y) resolution than traditional core turbulence simulations, they are characteristic of relatively low-resolution multiscale cases. The raw node-based results for `nl03` are plotted in Fig. 2a. The same timing data, but normalized to peak hardware performance results, are presented in

Table 3: Architecture overview of CPU-only systems including theoretical peak. TFLOP/node will be used as a normalizing factor in the performance analysis. Stampede2 Skylake nodes are hereafter referred to as Skylake.

	Cori	Stampede2	Skylake
Architecture	CPU	CPU	CPU
CPU Model	Xeon Phi 7250	Xeon Phi 7250	Xeon Plat 8160
Threads/node	272 (128 used)	272 (128 used)	96
TFLOP/node	3.0	3.0	3.5
Nodes	9668	4200	1736
Interconnect	Cray Aries	Intel Omni-Path	Intel Omni-Path
Net. Topology	Dragonfly	Fat Tree	Fat Tree
Compiler	Intel Fort 17	Intel Fort 17	Intel Fort 17
FFT library	FFTW v3.3.6	Intel MKL	Intel MKL
MPI	Cray MPICH v7.6	MPICH v7.6	MPICH v7.6

Fig. 2b. Generally speaking, since we do not presently instrument the actual hardware FLOP rate, we instead offer the peak hardware normalized results so that at least the relative closeness to vendor peak can be measured. Notable conclusions are that all systems scale well with an eventual degradation of performance at the highest core counts due to increasing problem granularity. The best performer is the TACC Stampede2 machine, which operates closest to the theoretical peak. Next are NERSC Cori and CSCS Piz Daint, which each achieve about the same fraction of theoretical peak, although the greater effective size of the Piz Daint system ultimately allows the shortest wallclock time. It is perhaps not surprising, given its age, that Titan achieves the lowest fraction of peak and overall performs less well than the other systems.

For the KNL systems, only 64 (of 68) cores per chip are used in order to facilitate problem splitting (i.e., array distribution). Moreover, we configure the runtime environment to use 2 out of 4 possible hyperthreads. Although in some cases we see a small performance improvement with the maximum 4 hyperthreads, the improvement is usually *insignificant*. All KNL nodes were configured to use the cache memory mode and quadrant cluster mode. We remark that there is an alternative MPI rank ordering method available in CGYRO that we do not use but offers performance improvements (reducing time spend in MPI communication) for many cases. It is suggested that users try the alternative scheme (`MPI_RANK_ORDER=2`) on a case-by-case basis. Understanding the per-system performance in more detail requires a deeper, kernel-level analysis as described in the next section.

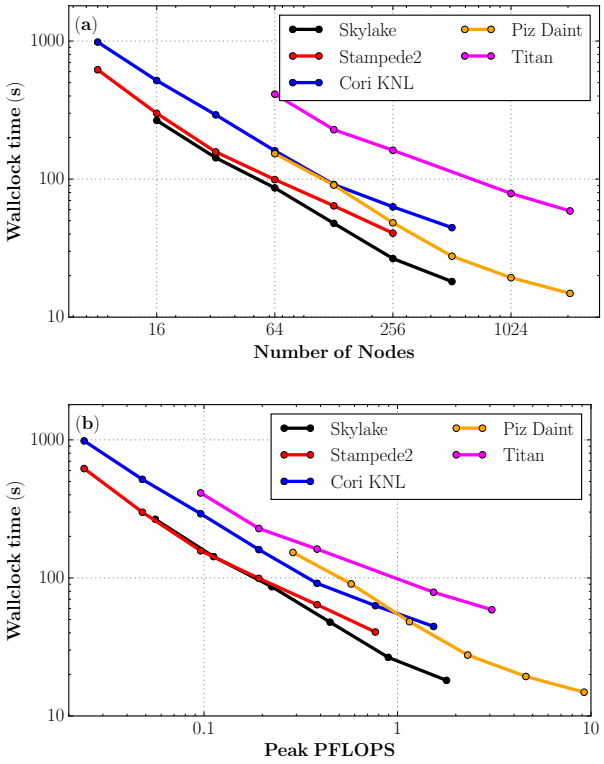

Figure 2: 5-platform strong-scaling comparison based on the CGYRO `nl03` test case, showing wallclock time versus (a) number of nodes, and (b) peak PFLOP rating. In (b), lower curves indicate performance closer to the vendor claim.

4.2. Kernel-based performance analysis

A breakdown of the time spent in each computational kernel, with the relevant kernels defined in Table 1, is summarized in Figs. 3a-c. Here, the comparison is normalized using three different approaches: (a) equal wallclock time, (b) equal nodes, (c) equal 0.2 peak PFLOP. In (a), the bar area is roughly constant, but slower systems need to use more hardware. In the latter two cases, shorter bars mean (b) better performance per node and (c) closer to peak performance rating. For clarity, we note that the normalization in (a) is defined by the intersection of the timing curves in Fig. 2 with the horizontal line $t = 100s$.

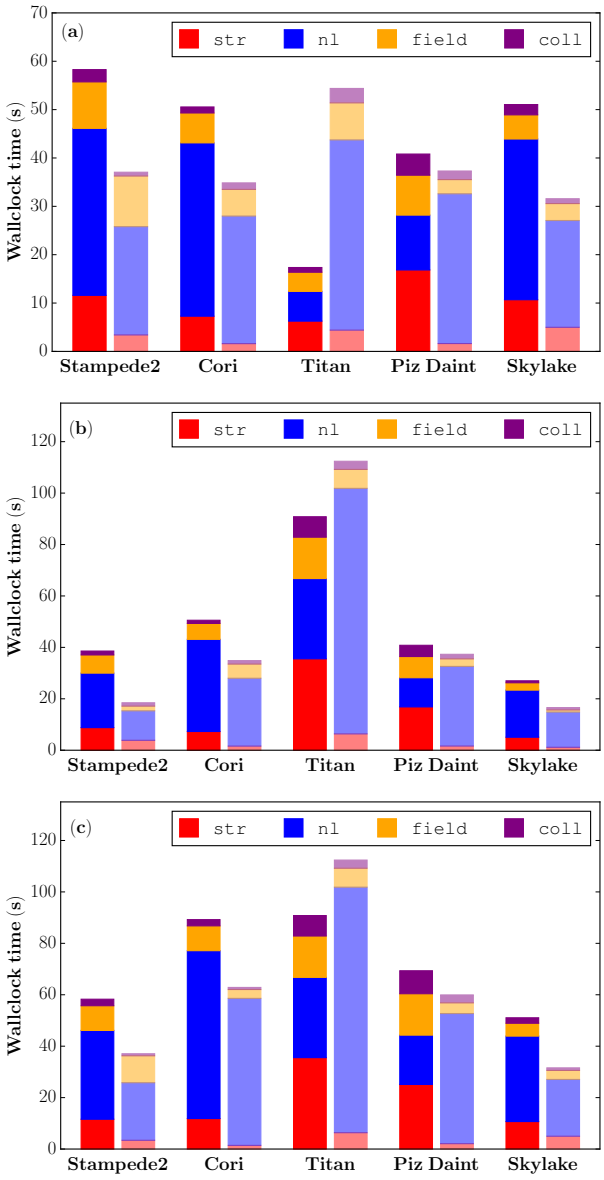

Figure 3: Kernel-level analysis of all 5 platforms for `nl03` , with data taken at (a) **fixed wallclock time**, (b) **fixed 128 nodes**, (c) **fixed peak 0.2 PFLOPS**. For each system, left bars indicate compute time, and right (slightly faded) bars are the corresponding communication (`_comm`) times. In (a), the equal-time metric ensures total bar area is roughly constant. However, this means the slower systems must use significantly more hardware than the faster system (e.g., Titan required 4x the number of nodes as Piz Daint for this metric). In (b), shorter bars indicate closer to peak performance rating. The interpretation of communication bars is more subtle since the number of nodes differs.

A number of conclusions are evident. First, on the CPU systems, the compute time is highly dominated by the nonlinear step. This is indeed a feature of the spectral algorithm that simpifies the linear dynamics, pushing the computational burden to the nonlinear term which is evaluated with a series of 2D FFTs as described in Section 3.6. On the GPU systems, the extremely high performance of cuFFT gives rise to a relatively short time spent in `nl`. This is evident in both Piz Daint and Titan. On the CPU systems, the time spent in `nl` is higher. For reference, the equivalent 1D size of the FFTs is about $n = 146$k for `nl03` . However, this value is by no means maximal. In the results section we show a production simulation of the DIII-D tokamak with $n = 587$k, and planned multiscale DIII-D ITER baseline simulations will have a formidable $n > 5$M. A second apparent feature of the kernel timings is the *high cost of the nonlinear communication*, `nl_comm`, which is implemented using `MPI_ALLTOALL` communication outside the FFT library. We emphasize that for the CPU systems, we use only the single-thread version of the FFT library.[1] On the CPU systems, the cost of `nl_comm` is always smaller than the cost of `nl`, whereas on the CPU/GPU systems the opposite is true. The nature of the algorithm unfortunately requires the movement of a significant amount of data for each timestep, so systems with a relatively high ratio of floating point to network performance are not ideal for CGYRO. At this point, a number of summary statements can be made.

- According to Figs. 3b and c, MKL FFT (Stampede2 KNL) significantly outperforms FFTW (Cori KNL).
- Regarding interconnects, according to Fig. 3b, we find that Intel Omni-Path outperforms Cray Aries, giving a very good balance with the floating-point performance on the Stampede2 systems.
- The nearly 8-year-old Cray Gemini interconnect (Titan) shows its age and is the poorest performer of the group.
- Regarding kernels, `nl` and `coll` tend to be *compute bound* on CPU systems and *memory bound* on GPU systems.
- The `str` and `field` kernels tend to be *compute bound*.
- The most expensive (typically) compute kernel, `nl`, appears to be well-optimized for both CPU and GPU.
- Future efforts for GPU systems will focus on (1) improving the GPU performance of the `str` kernel, and (2) porting the remaining `coll` and `field` kernels to GPU.

4.3. OpenMP scaling performance

In order to achieve maximal scalability, CGYRO employs a *hybrid MPI* approach. Broadly speaking, an efficient but coarse-grained problem spitting is made at the MPI level, with finer-grained problem splitting algorithms (OpenMP or GPU) inside compute kernels (see Section 3.7). This facilitates the use of accelerators (GPUs) and to better utilize the available CPU resources. As an example, by choosing 3 OpenMP threads on Skylake nodes, one can subsequently keep the number of MPI tasks as a power of 2.

[1] The large number of FFTs per MPI task mean that it is typically better to apply OpenMP to the loop over FFTs than to attempt multithreaded FFTs.

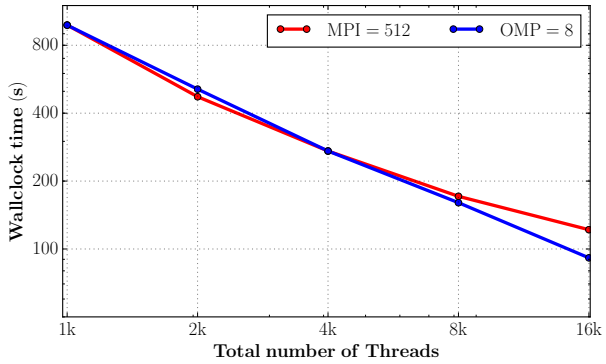

Figure 4: Hybrid MPI scaling for `nl03` on Cori KNL. The red curve shows scaling at fixed MPI task count (from 2 OMP threads on 8 nodes to 32 OMP threads on 128 nodes). The blue curve shows scaling at fixed OMP thread count (from 128 MPI tasks at 8 nodes to 2048 MPI tasks on 128 nodes). Thus there is a nearly a perfect tradeoff between OMP and MPI, except for the last OMP (red) scaling point (32 OMP threads).

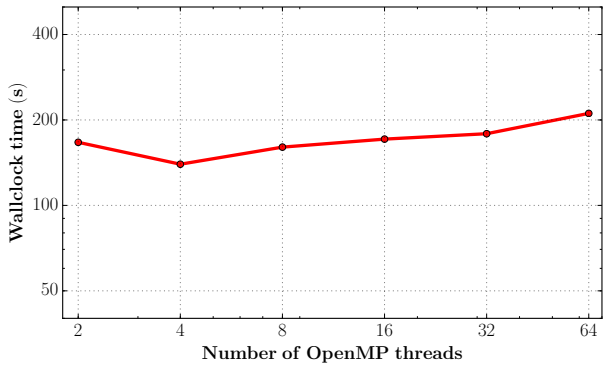

Figure 5: OpenMP strong scaling results for `nl03` on 64 Cori KNL nodes. The product of MPI tasks and OpenMP threads is fixed at 8192.

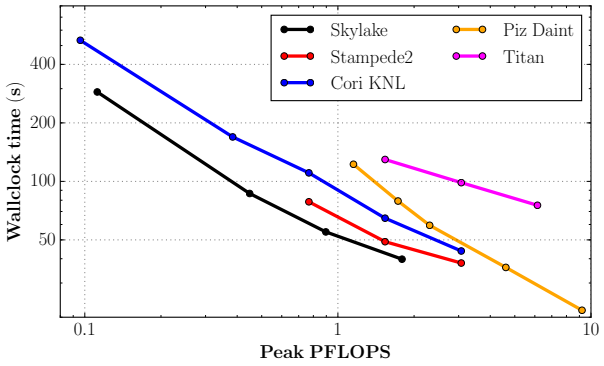

Figure 6: 5-platform strong-scaling comparison based on the CGYRO `nl04` test case, normalized by peak PFLOPS, to be compared with Fig. 2b for `nl03` .

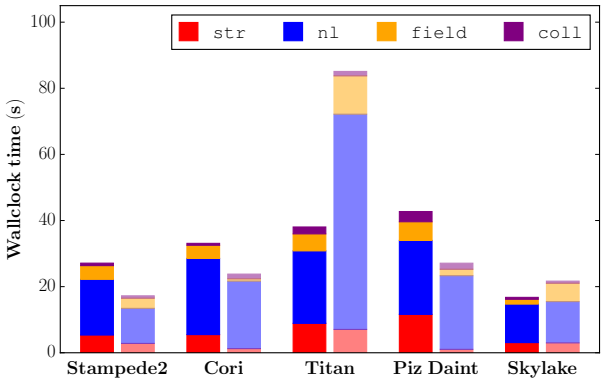

Figure 7: Same as Fig. 3c, except for case `nl04` taken at fixed peak 1.6 PFLOPS.

To showcase the effectiveness of this approach, in Fig. 4 we show the strong scaling plot of `nl03` on Cori, going from 1k to 16k threads. First, we vary the number of OpenMP threads, keeping the number of MPI tasks fixed (red curve). Next, we vary the number of MPI tasks, keeping the number of OpenMP threads fixed (blue curve). As can be seen, the measured timings are almost identical, with the highest thread counts giving a small advantage to MPI. As an alternative viewpoint, in Fig. 5 we present OpenMP thread scaling using a fixed amount of hardware. In order to have the widest possible range, we run the test on the high-thread-count KNL nodes (specifically, 64 Cori nodes each providing 128 hardware threads). The result is that OpenMP performance suffers minimal degradation even up to 64 threads per MPI task.

4.4. Performance analysis for larger case: nl04

In this section we apply the same performance analysis to a larger case, `nl04` , for which

$$(N_x, N_y, N_\theta, N_\xi, N_\mathrm{v}, N_a) = (512, 256, 48, 24, 8, 4) \ .$$

The equivalent 1D FFT size for the `nl` kernel is $n = 293\mathrm{k}$, and moreover 737k of these (forward and inverse) FFTs must be computed for each RK4 timestep. This truly underscores

the need for a high performance FFT implementation. Wall-clock time versus peak PFLOPS for `nl04` is shown in Fig. 6 (compare with Fig. 2). Here, we see a more clear separation between systems than for `nl03` . Stampede2 Skylake achieves the closest to the vendor-stated peak performance, followed by Stampede2 KNL, Cori KNL, Piz Daint and finally Titan. The kernel-level analysis is given in Fig. 7. Interestingly, the exceptional cuFFT performance on `nl03` is not as clearly evident for the `nl04` case. At this time, we have no clear explanation of this result.

5. Collisional, Electromagnetic Plasma Simulation

5.1. Need for an implicit collision scheme

Physically, the collision operator describes diffusion in each of the velocity-space dimensions. The necessity for using an implicit collision algorithm can be clearly illustrated by plotting the results of a linear simulation with a single value of k_y. In Fig. 8, we show a contour plot of the imaginary part of the electron distribution function in (θ, ξ) space, with each frame illustrating a separate energy meshpoint, v_i. Because the effective collision rate, $\nu_{ee}^D(\mathrm{v})$, diverges at low velocity, there is always a transition from a highly-collisional regime at low velocity to a nearly collisionless regime at the highest velocities. The islands apparent in frames $i = 5, 7$ of Fig. 8 represent the trapped-electron population. This population is progressively washed away (detrapped) by pitch-angle collisions, as shown in Eq. (20) for $i < 5$.

5.2. Nonlinear, electromagnetic multiscale simulation

In this section, we show results for a production multiscale simulation of a DIII-D plasma. The case (shot 164988) is an ITER baseline scenario discharge with plasma current $I_p = 1.2\,\mathrm{MA}$, toroidal field $B_T = -2.0\,\mathrm{T}$, beam heating power $P_{\mathrm{NB}} = 4\,\mathrm{MW}$, and small input torque [45]. It is typical of low-rotation DIII-D ITER baseline scenario discharges. We study the region centered at $r/a = 0.92$, and covered with a $26\rho_s \times 26\rho_s$ perpendicular domain. The wavenumber resolution covers electron gyroradius scales: $k_y\rho_s \le 32$ ($\Delta k_y\rho_s = 0.25$) and $k_x^0\rho_s \le 124$. This required $N(k_x^0) = 1024$ and $N(k_y) = 256$ (simulated with 128 complex modes). The most unstable linear eigenmode occurs at $k_y\rho_s = 0.5$. We remark that simulations with $\Delta k_y\rho_s = 0.5$ do not sufficiently resolve long wavelengths and are observed to be poorly behaved. Electric field shear was included using the wavenumber advection algorithm [36]. At the central radius, the total energy fluxes, as determined by experimental power balance analysis, are $Q_i/Q_{\mathrm{GB}} = 2.5$ and $Q_e/Q_{\mathrm{GB}} = 8.2$, where $Q_{\mathrm{GB}} \doteq n_e c_s T_e \rho_s^2/a^2 \simeq 4 \times 10^{-3}\,\mathrm{MW/m^2}$. Neoclassical calculations with NEO predict a neoclassical ion energy flux of $Q_i/Q_{\mathrm{GB}} = 2.7$, meaning that the ion transport in this discharge is *purely neoclassical*. The electron transport on the other hand cannot be accounted for by neoclassical effects. Estimates with the TGLF transport model indicate that the expected long-wavelength electron energy flux is $Q_e/Q_{\mathrm{GB}} < 0.4$. This suggests that nearly all of electron energy flux arises from short-wavelength turbulence, making this an ideal candidate discharge for multiscale analysis. And indeed, direct multiscale GK simulations predict $Q_e/Q_{\mathrm{GB}} \simeq 8$ with a broad spectrum that is peaked in the range $k_y\rho_s \simeq 8$ or $k_y\rho_e \simeq 0.13$, as shown in Fig. 9.

While it may appear that the k_x^0 range is over-resolved, this resolution is nevertheless required to maintain accuracy at the maximum k_y. The equivalent mesh spacing in real space is $\Delta x \simeq 1.5\rho_e$, which is somewhat smaller than the original rule-of-thumb $\Delta x = 2\rho_e$ for ETG simulation [46]. We can further reconstruct any desired moments of the distribution function via velocity integration and then interpolation onto an arbitrary radial mesh. Reconstructing perpendicular fluctuations at the outboard midplane $\theta = 0$ is particularly straightforward and can be done without reference to complicated geometric coefficients. For example, for the energy fluctuations, we use:

$$\delta E_a(x,y) = \sum_{k_x^0,k_y} \int d^3\mathrm{v}\, m_a \mathrm{v}^2 J_0(k_\perp \rho_a) \widetilde{H}_a\, e^{i(k_x^0 x + k_y y)}\,, \qquad (27)$$

with the integrand evaluated at $\theta = 0$. In Figs. 10 and 11, energy fluctuations are plotted in a plane perpendicular to the fieldline at $\theta = 0$. It is more complicated to reconstruct a toroidal cut because $y = y(\varphi, \theta)$ is a Clebsch (fieldline) angle [34], related in complicated way to the toroidal and poloidal angles. A detailed explanation requires the introduction of the plasma geometry and is beyond the scope of the present report.

6. Summary

In this report we have described the mathematical formulation, numerical discretization, and performance/scaling results for the new CGYRO gyrokinetic code. The performance data was collected on 5 current leadership systems (2 KNL-based, 2 hybrid CPU-GPU and 1 Skylake-based). For the cross-machine performance analysis, we compared timings for 8 separate computational kernels, thereby illustrating the strengths and weaknesses of the floating-point and communication architectures of the respective systems. Excellent strong scaling results are observed on both multicore-CPU and CPU/GPU systems. OpenMP scaling was demonstrated up to 64 OpenMP threads per MPI task. We concluded by showing new multiscale turbulence results in agreement with DIII-D experiments, recovering electron transport levels in an ITER-baseline plasma regime that cannot be described using traditional long-wavelength simulation.

7. Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, through the AToM project under Grant DE-SC0017992 and by the Edge Simulation Laboratory project under Grant DE-FC02-06ER54873. The research used resources of the Oak Ridge Leadership Computing Facility under Contract DE-AC05-00OR22725 and of the National Energy Research Scientific Computing Center under Contract No. DE-AC02-05CH11231. We also acknowledge support by the Swiss National Supercomputing Centre

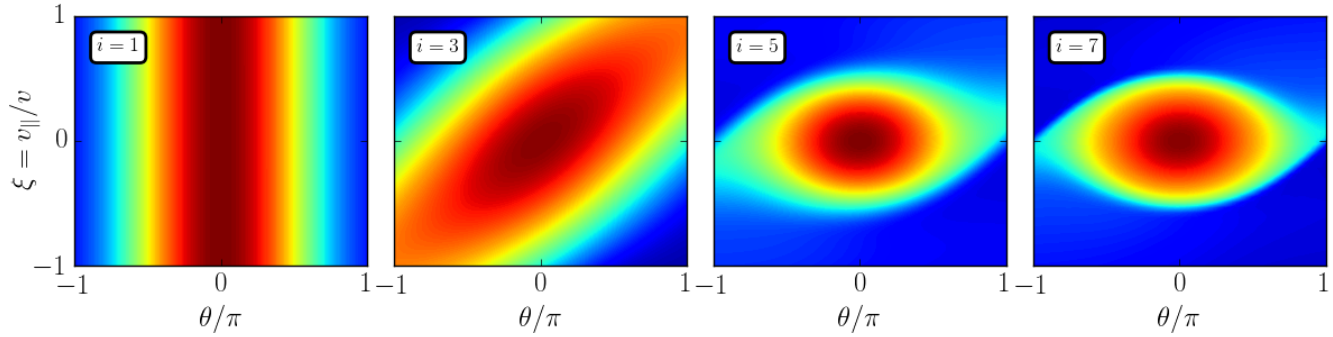

Figure 8: Contour plot of electron distribution function in (θ, ξ) space, with each frame illustrating a separate energy v. This is a linear simulation with a single value of k_y. There are multiple k_x^0 values, with the plot showing only $k_x^0 = 0$. The effective collision rate for each panel decreases strongly with energy.

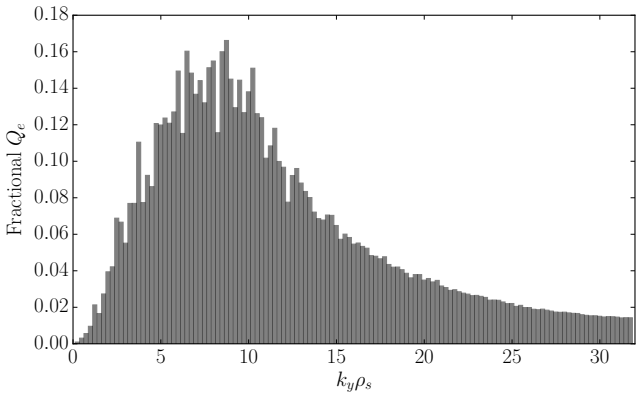

Figure 9: Fractional contribution to electron energy flux, Q_e/Q_{GB}, as a function of binormal wavenumber k_y. Total electron flux is equal to the area under the curve.

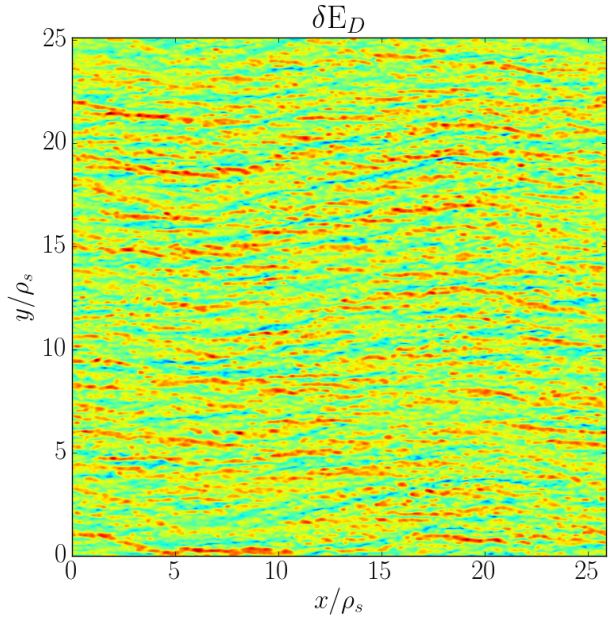

Figure 10: Deuterium energy fluctuations in a plane perpendicular to the fieldline at $\theta = 0$.

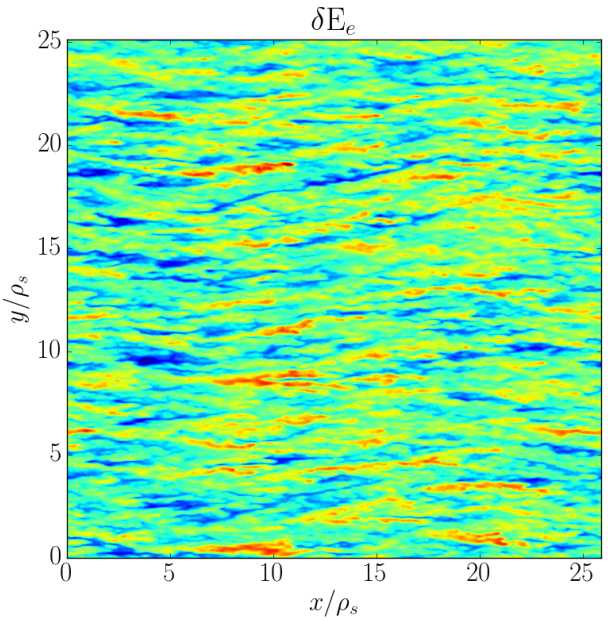

Figure 11: Electron energy fluctuations in a plane perpendicular to the fieldline at $\theta = 0$.

(CSCS) under project ID s819, and thank the Texas Advanced Computing Center (TACC) for providing HPC resources that contributed to this paper. In part, this work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053, and been supported by the EUROfusion High Performance Computer.

References

[1] H. Sugama, W. Horton, Neoclassical electron and ion transport in toroidally rotating plasmas, Phys. Plasmas 4 (1997) 2215.

[2] H. Sugama, W. Horton, Nonlinear electromagnetic gyrokinetic equation for plasmas with large mean flows, Phys. Plasmas 5 (1998) 2560.

[3] F. Hinton, R. Hazeltine, Rev. Mod. Phys. 48 (1976) 239.

[4] P. Helander, D. Sigmar, Collisional transport in magnetized plasmas, Cambridge University Press, Cambridge, 2002.

[5] E. Frieman, L. Chen, Nonlinear gyrokinetic equations for low-frequency electromagnetic waves in general plasma equilibria, Phys. Fluids 25 (1982) 502.

[6] J. Candy, C. Holland, R. Waltz, M. Fahey, E. Belli, Tokamak profile prediction using direct gyrokinetic and neoclassical simulation, Phys. Plasmas 16 (2009) 060704.

[7] O. Meneghini, P. B. Snyder, S. P. Smith, J. Candy, G. M. Staebler, E. A. Belli, L. L. Lao, J. M. Park, D. L. Green, W. Elwasif, B. A. Grierson, C. Holland, Integrated fusion simulation with self-consistent core-pedestal coupling, Phys. Plasmas 23 (2016) 042507.

[8] W. Horton, Drift waves and transport, Rev. Mod. Phys. 71 (1999) 735.

[9] G. Staebler, J. Kinsey, R. Waltz, A theory-based transport model with comprehensive physics, Phys. Plasmas 14 (2007) 055909.

[10] C. Bourdelle, J. Citrin, B. Baiocchi, A. Casati, P. Cottier, X. Garbet, F. Imbeaux, et al., Core turbulent transport in tokamak plasmas: bridging theory and experiment with QuaLiKiz, Plasma Phys. Control. Fusion 58 (2016) 014036.

[11] ITER Physics Basis Editors, et al., Nucl. Fusion 39 (1999) 2137.

[12] D. Campbell, Preface to special topic: ITER, Phys. Plasmas 22 (2015) 021701.

[13] C. Holland, A. White, G. McKee, M. Shafer, J. Candy, R. Waltz, L. Schmitz, G. Tynan, Implementation and application of two synthetic diagnostics for validating simulations of core tokamak turbulence, Phys. Plasmas 16 (2009) 052301.

[14] A. Casati, T. Gerbaud, P. Hennequin, C. Bourdelle, J. Candy, F. Clairet, X. Garbet, V. Grandgirard, O. Gurcan, S. Heuraux, G. Hoang, C. Honoré, F. Imbeaux, R. Sabot, Y. Sarazin, L. Vermare, R. Waltz, Turbulence in the TORE SUPRA tokamak: Measurements and validation of nonlinear simulations, Phys. Rev. Lett. 102 (2009) 165005.

[15] C. Holland, L. Schmitz, T. Rhodes, W. Peebles, J. Hillesheim, G. Wang, L. Zeng, E. Doyle, S. Smith, R. Prater, K. Burrell, J. Candy, R. Waltz, J. Kinsey, G. Staebler, J. DeBoo, C. Petty, G. McKee, Z. Yan, A. White, Advances in validating gyrokinetic turbulence simulations against L- and H-mode plasmas, Phys. Plasmas 18 (2011) 056113.

[16] C. Holland, C. Petty, L. Schmitz, K. Burrell, G. McKee, T. Rhodes, J. Candy, Progress in GYRO validation studies of DIII-D H-mode plasmas, Nucl. Fusion 52 (2012) 114007.

[17] L. Schmitz, C. Holland, T. Rhodes, G. Wang, L. Zeng, A. White, J. Hillesheim, W. Peebles, S. Smith, R. Prater, G. McKee, Z. Yan, W. Solomon, K. Burrell, C. Holcomb, E. Doyle, J. DeBoo, M. Austin, J. deGrassie, C. Petty, Reduced electron thermal transport in low collisionality H-mode plasmas in DIII-D and the importance of TEM/ETG-scale turbulence, Nucl. Fusion 52 (2012) 023003.

[18] F. Casson, R. McDermott, C. Angioni, et al., Validation of gyrokinetic modelling of light impurity transport including rotation in ASDEX upgrade, Nucl. Fusion 53 (2013) 063026.

[19] J. Citrin, F. Jenko, P. Mantica, D. Told, C. Bourdelle, R. Dumont, J. Garcia, J. Haverkort, G. Hogeweij, T. Johnson, M. Pueschel, J.-E. contributors, Ion temperature profile stiffness: non-linear gyrokinetic simulations and comparison with experiment, Nucl. Fusion 54 (2014) 023008.
URL `http://stacks.iop.org/0029-5515/54/i=2/a=023008`

[20] C. Sung, A. White, D. Mikkelsen, M. Greenwald, C. Holland4, N. Howard, R. Churchill, C. Theiler, A. C.-M. Team, Quantitative comparison of electron temperature fluctuations to nonlinear gyrokinetic simulations in C-Mod Ohmic L-mode discharges, Phys. Plasmas 23 (2016) 042303.

[21] A. White, W. Peebles, T. Rhodes, C. Holland, G. Wang, L. Schmitz, T. Carter, J. Hillesheim, E. Doyle, L. Zeng, et al., Measurements of the cross-phase angle between density and electron temperature fluctuations and comparison with gyrokinetic simulations), Physics of Plasmas 17 (5) (2010) 056103.

[22] J. Candy, R. Waltz, M. Fahey, C. Holland, The effect of ion-scale dynamics on electron-temperature-gradient turbulence, Plasma Phys. Control. Fusion 49 (2007) 1209.

[23] N. Howard, C. Holland, A. White, M. Greenwald, J. Candy, A. Creely, Multi-scale gyrokinetic simulations: Comparison with experiment and implications for predicting turbulence and transport, Phys. Plasmas 23 (2016) 056109.

[24] N. Howard, C. Holland, A. White, M. Greenwald, J. Candy, Multi-scale gyrokinetic simulation of tokamak plasmas: enhanced heat loss due to cross-scale coupling of plasma turbulence, Nucl. Fusion 56 (1) (2016) 014004.
URL `http://stacks.iop.org/0029-5515/56/i=1/a=014004`

[25] C. Holland, N. Howard, B. Grierson, Gyrokinetic predictions of multiscale transport in a DIII-D ITER baseline discharge, Nucl. Fusion 57 (2017) 066043.

[26] N. Howard, C. Holland, A. White, M. Greenwald, P. Rodriguez-Fernandez, J. Candy, A. Creely, Multi-scale gyrokinetic simulations of an Alcator C-Mod, ELM-y H-mode plasma, Plasma Phys. Control. Fusion 60 (2017) 014034.

[27] J. Candy, R. Waltz, An Eulerian gyrokinetic-Maxwell solver, J. Comput. Phys. 186 (2003) 545.

[28] J. Candy, R. Waltz, Anomalous transport in the DIII-D tokamak matched by supercomputer simulation, Phys. Rev. Lett. 91 (2003) 045001–1.

[29] J. Candy, R. Waltz, W. Dorland, The local limit of global gyrokinetic simulations, Phys. Plasmas 11 (2004) L25.

[30] J. Candy, E. Belli, GYRO Technical Guide, General Atomics Technical Report GA-A26818.

[31] J. Kinsey, R. Waltz, J. Candy, Nonlinear gyrokinetic turbulence simulations of E×B shear quenching of transport, Phys. Plasmas 12 (2005) 062302.

[32] J. Kinsey, R. Waltz, J. Candy, The effect of safety factor and magnetic shear on turbulent transport in nonlinear gyrokinetic simulations, Phys. Plasmas 13 (2006) 022305.

[33] J. Kinsey, R. Waltz, J. Candy, The effect of plasma shaping on turbulent transport and E×B shear quenching in nonlinear gyrokinetic simulations, Phys. Plasmas 14 (2007) 102306.

[34] J. Candy, E. Belli, R. Bravenec, A high-accuracy Eulerian gyrokinetic solver for collisional plasmas, J. Comput. Phys. 324 (2016) 73.

[35] E. Belli, J. Candy, Implications of advanced collision operators for gyrokinetic simulation, Plasma Phys. Control. Fusion 59 (2017) 045005.

[36] J. Candy, E. Belli, Spectral treatment of gyrokinetic shear flow, J. Comput. Phys. 356 (2018) 448.

[37] E. Belli, J. Candy, Impact of centrifugal drifts on ion turbulent transport, Phys. Plasmas 25 (2018) 032301.

[38] E. Belli, J. Candy, Kinetic calculation of neoclassical transport including self-consistent electron and impurity dynamics, Plasma Phys. Control. Fusion 50 (2008) 095010.

[39] E. Belli, J. Candy, Full linearized Fokker-Planck collisions in neoclassical transport simulations, Plasma Phys. Control. Fusion 54 (2012) 015015.

[40] The General Atomics GACODE Suite, `http://gafusion.github.io/doc` (2018).

[41] E. Belli, J. Candy, Fully electromagnetic gyrokinetic eigenmode analysis of high-beta shaped plasmas, Phys. Plasmas 17 (2010) 112314.

[42] R. Hager, J. Lang, C. Chang, S. Ku, Y. Chen, S. Parker, M. Adams, Verification of long wavelength electromagnetic modes with a gyrokinetic-fluid hybrid model in the XGC code, Phys. Plasmas 24 (2017) 054508.

[43] S. Orszag, On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components, J. Atmos. Sci. 28 (1971) 1074.

[44] M. Frigo, S. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2) (2005) 216.

[45] S. Haskey, B. Grierson, et al., Main ion and impurity profile evolution across the L- to H-mode transition on DIII-D, Plasma Phys. Control. Fusion 60 (2018) 105001.

[46] W. Nevins, J. Candy, S. Cowley, T. Dannert, A. Dimits, W. Dorland, C. Estrada-Mila, G. Hammett, F. Jenko, M. Pueschel, D. Shumaker, Characterizing electron temperature gradient turbulence via numerical simulation, Phys. Plasmas 13 (2006) 122306.