

Block-Structured Grids for Eulerian Gyrokinetic Simulations

D. Jarema^{a,*}, H.J. Bungartz^a, T. Görler^b, F. Jenko^{c,b}, T. Neckel^a, D. Told^{c,b}

^a*Institut für Informatik, Technische Universität München, Boltzmannstraße 3, 85748 Garching, Germany*

^b*Max-Planck-Institut für Plasmaphysik, Boltzmannstraße 2, 85748 Garching, Germany*

^c*Department of Physics and Astronomy, University of California, Los Angeles, CA 90095, USA*

Abstract

In order to predict the turbulent transport in magnetic fusion experiments, global (i.e., full-torus) gyrokinetic simulations are often carried out. In this context, one frequently encounters situations in which the plasma temperature varies by large factors across the radial simulation domain. In grid-based Eulerian codes, this enforces the use of a very large number of grid points in the two-dimensional velocity space, and, thus, an enormous computational effort. To minimize the computational requirements, one may employ block-structured grids, adapted to the radial changes of the temperature. As the block-structured grids rely on a general approach, they can be applied to different Eulerian gyrokinetic implementations. In this paper, we explain the construction and implementation of such grids in the gyrokinetic code GENE, F. Jenko et al. (2000), and present corresponding simulation results.

Keywords: Plasma turbulence, gyrokinetic simulation, block-structured grids

1. Introduction

Plasma microturbulence is responsible for causing strong outward transport of heat and particles in fusion experiments, making a thorough understanding of it of paramount importance to the development of future nuclear fusion reactors. Gyrokinetics has been shown to be an appropriate model to simulate microturbulence in magnetically confined core plasmas [1, 2]. In spite of a reduction from six to five dimensions and a removal of irrelevant space-time scales compared to the full kinetic description, gyrokinetic simulations are still computationally expensive. For example, an estimated grid for the future large-scale

*Corresponding author

fusion experiment ITER [3] simulations contains around ten billion grid points¹. Furthermore, recent studies indicate that, depending on the temperature and density profiles, simulations may require a significantly higher number of grid points, even up to a trillion. This amount of degrees of freedom is, in contrast to other applications, computationally prohibitively expensive for already fully parallelized gyrokinetic simulations, imposing the need for methods that allow reducing considerably the number of grid points without a significant loss of accuracy.

In this paper, we discuss the use of block-structured grids in the Eulerian gyrokinetic code GENE (Gyrokinetic Electromagnetic Numerical Experiment) [4, 5, 6], resulting in a significant reduction in the number of grid points and, thus, an improvement in the performance of radially extended gyrokinetic simulations, while at the same time maintaining accuracy. Furthermore, the design and development of block-structured grids integrate smoothly in the existing implementation, since the block-structured grids use primarily the same routines as the regular grids, after certain adaptations.

GENE solves a set of nonlinear integro-differential equations (coupled to Maxwell's equations) for the particle distribution functions in a five-dimensional position-velocity phase space. GENE is based on the method of lines, discretizing phase space and time separately. The time stepping follows a fourth-order explicit Runge-Kutta method. GENE's implementation includes several well-established techniques addressing the mathematical model to reduce the computational demands. First, a δf -splitting is applied where the fluctuating part is considered small with respect to the equilibrium part. Furthermore, a field-aligned coordinate system is employed to exploit the strong anisotropy of turbulent fluctuations in the directions parallel and perpendicular to the background magnetic field [7, 8]. Finally, the spatial grid is adapted to the background magnetic field. Details on the mathematical model and implementation are given in [9, 10, 11].

In GENE, there are two main operation modes, local and global, each of them with its own requirements on computational grids. In the local simulations [12], the computational domain follows the magnetic field line, and its radial size is small compared to the fusion device size. Here, the radial variation of the temperature and density profiles can be neglected. In this case, the two spatial directions perpendicular to the background magnetic field are treated using efficient spectral methods, whereas the third spatial direction — parallel to the background magnetic field — and the velocity space directions are discretized via finite difference methods.

In the global operation mode, however, where the radial changes of the plasma parameters are of interest, neglecting the radial temperature and density profiles is no longer possible, because the grid length in radial direction may

¹Number of grid points in each direction: `n_spec` = 2, `nx0` = 2048, `nky0` = 32, `nz0` = 24, `nv0` = 96, `nw0` = 32. The computational grid does not resolve electron spatial scales. The coordinate system is described in Section 3.

extend up to the full machine size. Furthermore, the different types of boundary conditions that are employed in the radial direction prohibit the use of spectral methods in this dimension. Hence, finite element interpolation techniques are required to perform the gyro-averaging and construct the field solver. Regarding the computational grids in the global simulations, the straightforward approach is to extend the equidistant regular grids of the local simulations to the full radial range. This leads to a significant increase in the number of degrees of freedom in the velocity space, and makes certain global simulations computationally prohibitively expensive (details on velocity space resolution are available in [13, 14, 15, 16]). A method to alleviate this issue is by coupling multiple local simulations (cf., for example, the gyrokinetic codes TRINITY and TGYRO [17, 18]). However, as soon as the turbulence correlation length approaches the scale lengths of the background density and temperature profiles, global simulations are necessary.

The block-structured grids introduced in this paper address the problem of reducing the computational demands of grids in the global operation mode of GENE. We adapt the block-structured grids to the Maxwellian background particle distribution function, which depends on the temperature profile. The resulting computational grids are accurate and have considerably less nodes in the velocity space in comparison to the original regular grids. They also allow the reuse of a large amount of regular grid-specific code in GENE.

The rest of the paper is structured as follows. In Section 2, we introduce the governing equations solved in GENE, which are relevant for the grid construction. Section 3 presents the initial equidistant grids used in GENE, as well as the block-structured grids that can be deduced from them. Further, in Section 4, we explain the implementation of the proposed grids. The results are presented in Section 5. Finally, we summarize the contribution of this work in Section 6.

2. The gyrokinetic system of equations

This section intends to provide a brief introduction to gyrokinetic theory and an overview of the system of equations which is solved in gyrokinetic codes like GENE. Generally, hot and dilute plasmas cannot be treated as fluids since they do not thermalize at such low collisionalities and kinetic effects have to be retained. The theoretical framework of choice is hence given by the Boltzmann equation or — in the collisionless limit — by the Vlasov equation. Here, a distribution function $F_s(\mathbf{x}, \mathbf{v}, t)$ of a species s (electrons and fully ionized ions) in a six-dimensional space is propagated in time via

$$\frac{\partial F_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} F_s + q_s (\mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} F_s = 0, \quad (1)$$

where q_s is the charge of the species s , \mathbf{E} and \mathbf{B} are the electric and magnetic fields, respectively. These fields are computed consistently from the Maxwell equations, which, when formulated in the nonrelativistic limit for the electric

and magnetic potentials ϕ and \mathbf{A} , take the form

$$-\nabla^2 \phi = 4\pi \sum_s q_s n_s = 4\pi \sum_s q_s \int d^3v F_s, \quad (2)$$

$$-\nabla^2 \mathbf{A} = \frac{4\pi}{c} \sum_s q_s n_s \mathbf{u}_s = \frac{4\pi}{c} \sum_s q_s \int d^3v \mathbf{v} F_s, \quad (3)$$

where Coulomb gauge is used, n_s, \mathbf{u}_s are the density, and velocity of the species s , Eq. (2) and (3) being moments of the distribution function F_s .

Because of the high dimensionality, this set of equations is very expensive to simulate. However, since the gyration of charged particles in highly magnetized plasmas is typically much faster than the dynamics of interest, a reduced 5D description can be employed where the exact orbit information is erased and effectively only gyrorings are considered. Mathematically, this can be achieved by introducing gyroaverages, $\bar{A}(\mathbf{X}) \equiv \frac{1}{2\pi} \oint d\theta A(\mathbf{X} + \mathbf{r}(\theta))$, along the gyrophase θ and applying proper near-identity (Lie) transformations, for details, see [1, 19]. This yields a full- F five-dimensional gyrokinetic equation per species s :

$$\frac{\partial F_s}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla F_s + \frac{dv_{\parallel}}{dt} \frac{\partial F_s}{\partial v_{\parallel}} + \frac{d\mu}{dt} \frac{\partial F_s}{\partial \mu} = 0. \quad (4)$$

with the gyrocenter coordinate \mathbf{X} , the velocity component parallel to the magnetic field v_{\parallel} , and the magnetic moment $\mu = mv_{\perp}^2/2B$. In the electrostatic limit (for the sake of simplicity) and taking only the first order terms in the gyrokinetic expansion, their time evolutions are given by

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= v_{\parallel} \mathbf{b}_0 + \frac{B_0}{B_{0\parallel}^*} (\mathbf{v}_E + \mathbf{v}_{\nabla B} + \mathbf{v}_c), \\ \frac{dv_{\parallel}}{dt} &= -\frac{d\mathbf{X}/dt}{m_s v_{\parallel}} \cdot (q_s \nabla \bar{\phi}_1 + \mu \nabla B_0), \\ \frac{d\mu}{dt} &= 0. \end{aligned} \quad (5)$$

Here, B_0 denotes the modulus of the magnetic field vector \mathbf{B}_0 , $\mathbf{b}_0 = \mathbf{B}_0/B_0$ the corresponding unit vector, $B_{0\parallel}^* = \mathbf{b} \cdot \mathbf{B}_0^*$ the parallel component of $\mathbf{B}_0^* = \mathbf{B}_0 + \nabla \times (\mathbf{B}_0 v_{\parallel}/\Omega_s)$, and $\Omega_s = q_s B_0/m_s c$ is the gyrofrequency of species s with mass m_s . Furthermore, three characteristic drift terms appear: the $\mathbf{E} \times \mathbf{B}$ velocity $\mathbf{v}_E = \frac{c}{B_0^2} \mathbf{B}_0 \times \nabla \bar{\phi}$, the gradient-B drift $\mathbf{v}_{\nabla B_0} = \frac{\mu c}{q_s B_0^2} \mathbf{B}_0 \times \nabla B_0$, and the curvature drift velocity $\mathbf{v}_c = \frac{v_{\parallel}^2}{\Omega_s} (\nabla \times \mathbf{b}_0)_{\perp}$.

In so-called δf -codes like GEN \bar{E} , the gyrokinetic Vlasov equation is further simplified by splitting the full distribution function into a local Maxwellian background F_{0s} and a fluctuating part f_{1s} (see [20, 21]), which, according to the gyrokinetic ordering (see [1]), is by one order smaller than F_{0s} ($f_{1s}/F_{0s} \sim \epsilon$).

The local equilibrium distribution function F_{0s} depends on the density $n_s(x)$

and temperature $T_s(x)$ radial profiles, and is given by

$$F_{0s}(x, v_{\parallel}, \mu) = \frac{n_s(x)}{\pi^{3/2} v_T^3(x)} \exp \left[-\frac{mv_{\parallel}^2/2 + \mu B}{T_s(x)} \right]. \quad (6)$$

With this choice, the inherent problem motivating this work becomes obvious. The velocity space structures follow the thermal velocity $v_T(x) = \sqrt{2T_s(x)/m_s}$ and are hence typically large in the plasma core and small close to the edge. These various scales need to be resolved in the corresponding numerical grids. Here, moments of the equilibrium distribution function may help to define the minimum extent of the velocity space boxes. For instance, an accuracy limit may be set up to which the velocity space integration of $F_{0s}(x)$ should match the original density profile $n_s(x)$. However, some safety factor should be taken into account in order to accommodate contributions from the fluctuation part. At this point, it should be emphasized that this problem is not specific to δf -codes but is met in all grid-based models like full- F gyrokinetic or even kinetic codes due to the underlying physics issue.

In the implementation to be considered in the following, the δf code GENE, some further optimizations of the underlying equations are performed. First, only the lowest order nonlinearity, the so-called $E \times B$ nonlinearity is retained. Some higher order terms like the parallel nonlinearity can be switched on for testing but are usually found to be insignificant, see, e.g., [22]. Second, curvilinear field-aligned coordinates are used to exploit the high anisotropy of plasma turbulence and avoid some derivatives along the magnetic field lines. Since the details are not of direct relevance for the following discussion, the reader is referred to, e.g., [9, 10, 11] for the final version of the resulting system equation that furthermore includes collisions and electromagnetic effects.

3. Computational grids for gyrokinetic simulations

3.1. Regular computational grids

Fluctuations in gyrokinetic simulations are of a strongly anisotropic character, where parallel correlation lengths exceed the perpendicular lengths by several orders of magnitude. Therefore, field-aligned coordinates are employed for the positional space in GENE and other plasma simulation codes, which reduces the number of grid points compared to conventional grids. General information about field-aligned coordinates is available in [23], and details on the grid geometries implemented in GENE are provided in [8].

We use (x, y, z) symbols for the field-aligned spatial coordinates, where x is the radial coordinate or the flux surface label, y the binormal coordinate or field line label on a given flux surface, and z the parallel coordinate. The boundary conditions determine the discretization applied to the different directions. For instance, periodic boundary conditions in the y direction allow switching to the Fourier representation of this coordinate and using fast spectral methods. The parallel direction z is quasi-periodic ($f(x, k_y, z + L_z) =$

$f(x, k_y, z) \exp[-2\pi i n_0 q(x) j]$, where $q(x)$ is the safety factor, n_0 — an integer fraction of the toroidal turn for which the periodicity is assumed, and $k_y = j k_{y,\min}$ gives the toroidal mode number). A finite difference method on an equidistant grid is employed for this coordinate. A finite difference scheme is also applied to the radial coordinate x , due to the Dirichlet boundary condition used in the global simulations.

The complex curvilinear coordinate system in the position space is reflected by the metric coefficients and Jacobian, which appear in the computation of the derivatives and integrals of the gyrokinetic Vlasov-Maxwell equations. The logical or computational grid for the coordinates (x, y, z) is nevertheless a simple rectangular mesh with constant mesh step-sizes in all directions.

In the two-dimensional velocity space, we use the parallel velocity v_{\parallel} and the magnetic moment μ as coordinates. An equidistant grid is used for the discretization of the parallel velocity coordinate. For the μ direction, however, no derivatives have to be taken into account in collisionless plasma simulations, see Eq. (5). Only integrations are performed along this coordinate in order to compute the electromagnetic fields and for diagnostic purposes. Hence, for efficiency, Gauss-Laguerre knots are employed for the magnetic moment coordinate discretization.

The information on the default or regular grids in GENE is summarized in Table 1.

Table 1: GENE global version coordinates

symbol	name	space type	discretization
x	radial direction	position-configuration	equidistant
y, k_y	binormal direction	position-Fourier	equidistant
z	parallel direction	position-configuration	equidistant
v_{\parallel}	parallel velocity	velocity-configuration	equidistant
μ	magnetic moment	velocity-configuration	Gauss-Laguerre

3.2. Block-structured computational grids

In plasma simulations, block-structured grids are often used in the context of adaptive mesh refinement. An example of this approach for a Vlasov two-dimensional simulation (1D+1V — positional and velocity coordinates) is provided in [24], while another example implementing Particle-In-Cell (PIC) simulations is described in [25]. Locally refined block-structured grids are also applied in fluid plasma simulations, see [26, 27]. Other types of block-structured grids include the multiple connected blocks of grids in the logical computational domain, where each block corresponds to a physical simulation subdomain. An example of this type of grids for edge plasma simulations is described in [28]. The block-structured grids described in this paper are related to overlapping or overset grids [29, 30], because the data-exchange between blocks in these grids is done through interpolation in the overlapping region. The overset grids are often used to represent complex geometries and discretize regions with different

physical properties. In our case, the geometric shape of the simulation domain in the velocity space is not of interest. However, one should carefully choose the range and resolution for the velocity directions, depending on the temperature at the given radial distance.

This subsection describes the rationale for the construction of block-structured grids. We are interested in the fluctuating part of the distribution function, which typically has a structure similar to the equilibrium distribution function (6). For this reason, we use the Maxwellian distribution as the main factor that influences the shape and resolution of the grid.

The grid is built independently in different subspaces. We start with the radial distance x and parallel velocity v_{\parallel} subspace. To obtain the distribution function only in this subspace, we marginalize out (discard) all perpendicular velocity \mathbf{v}_{\perp} components by integrating the full Maxwellian over the $v_{\perp x}$ and $v_{\perp y}$ variables. This leads to

$$F_0(x, v_{\parallel}) = n(x) \sqrt{\frac{m}{2\pi T(x)}} \exp\left[-\frac{mv_{\parallel}^2}{2T(x)}\right]. \quad (7)$$

Note that in the last equation, without loss of generality, we omitted the species index s . The subgrids in the parallel velocity v_{\parallel} – radial distance x subspace can be constructed separately for different species.

The distribution function in the radial distance x and in the perpendicular velocity \mathbf{v}_{\perp} subspace is obtained similarly

$$F_0(x, v_{\perp x}, v_{\perp y}) = n(x) \frac{m}{2\pi T(x)} \exp\left[-\frac{m(v_{\perp x}^2 + v_{\perp y}^2)}{2T(x)}\right]. \quad (8)$$

In the five-dimensional gyrokinetic space, however, we work with the magnetic moment rather than with separate perpendicular velocity components:

$$\mu = \frac{m(v_{\perp x}^2 + v_{\perp y}^2)}{2B}. \quad (9)$$

The distribution function in the x and μ plane takes the form of an exponential distribution

$$F_0(x, \mu) = n(x) \frac{B}{T(x)} \exp\left[-\frac{\mu B}{T(x)}\right] = n(x) \lambda e^{-\lambda \mu}, \quad (10)$$

with the rate $\lambda = B/T(x)$.

The dependence of the background distribution functions (7) and (10) on the radial distance is transmitted via the density and temperature profiles $n(x)$ and $T(x)$. These profiles can be analytically generated or measurements can be taken directly from a fusion device, e.g., see in Figure 1 the electron temperature and density profiles of a particular “Tokamak à Configuration Variable” (TCV) discharge described in [31]. In the figure, the radial distance is shown in the

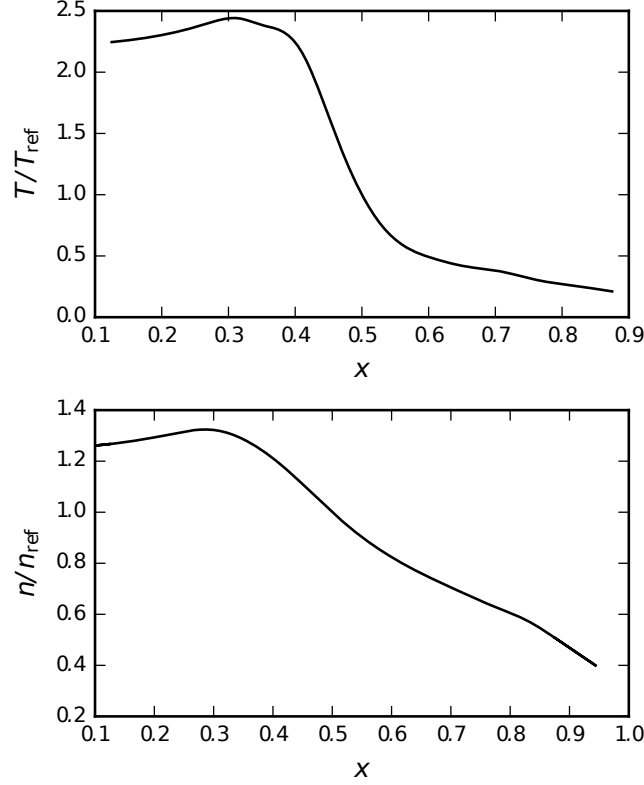


Figure 1: TCV electron temperature (top) and densities (bottom) profiles. The radial distance is shown in the minor radius units, the temperature and density values are shown in units relative to the reference values T_{ref} and n_{ref} taken at the reference radial distance ($x_{\text{ref}} = 0.5$).

minor radius units, the temperature and density values are shown relative to the reference values T_{ref} and n_{ref} taken at the reference radial distance, which in the particular example is $x_{\text{ref}} = 0.5$.

In the numerical simulations, we cannot take the full parallel velocity range from $-\infty$ to ∞ and the magnetic moment from 0 to ∞ . We thus restrict our parallel velocity direction to the range $[-1v, 1v]$ and the magnetic moment direction to $[0, 1w]$. The lengths of the ranges $1v$ and $1w$ are chosen based on experience and usually are bigger than 99% confidence intervals of the respective distribution functions. The confidence intervals of 99% can be considered as good starting values if good ranges are unknown.

In the local simulations, the temperatures and densities are fixed and taken from the profiles at the radial surface being studied. Therefore, to compute the confidence intervals, we only need the univariate distributions resulting from (7) and (10) at the fixed radial position.

For the global simulations, we cannot ignore the temperature and density

dependence on the radial distance. Figure 2 shows the distribution functions with corresponding contours of a 99% confidence interval for the $v_{\parallel} - x$ and $\mu - x$ subspaces. The confidence intervals and corresponding contours do not depend

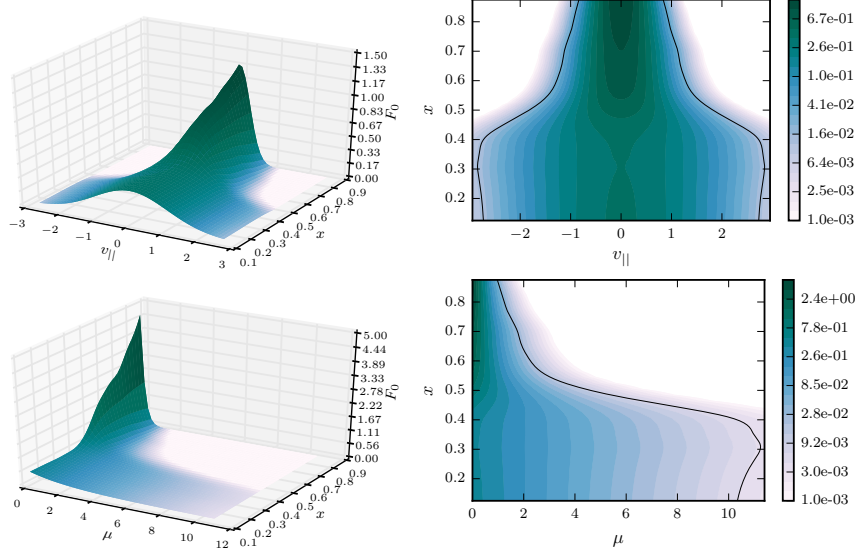


Figure 2: Background distribution functions in $v_{\parallel} - x$ and $\mu - x$ subspaces with corresponding confidence interval contours of 99%.

on the density profile, but on the temperature profile and radial variation in the magnetic field strength. It can be observed that the lengths of the parallel velocity and magnetic moment ranges vary strongly with the radial distance. For the global regular grids described in the previous subsection, the natural choice for lv and lw are the biggest ranges that correspond to the highest temperature of the whole simulation domain. The resulting grid has a lot of nodes that have negligible contribution to the simulation results.

We use the block-structured approximation (see the solid line in Figure 3) of the confidence interval contour to cut away nodes having a low equilibrium probability value from the initial regular grid. The resulting grids (shown in Figure 3) will be referred to, in the following, as the first type of block-structured grids.

The number of points that can be neglected depends on the given profile. The reduction of the grid nodes for one of the TCV profile linear simulation examples is displayed in Table 2. The second column (“regular”) of the table provides the number of points in the full regular grid. The third column (“BS 1”) shows how many points are left in the first type block-structured grid with six blocks. From the third column we observe that around 62% of the points from the regular grid are left in the final block-structured grid of the first type. Even if we cut away approximately half of the simulation domain in the $\mu - x$

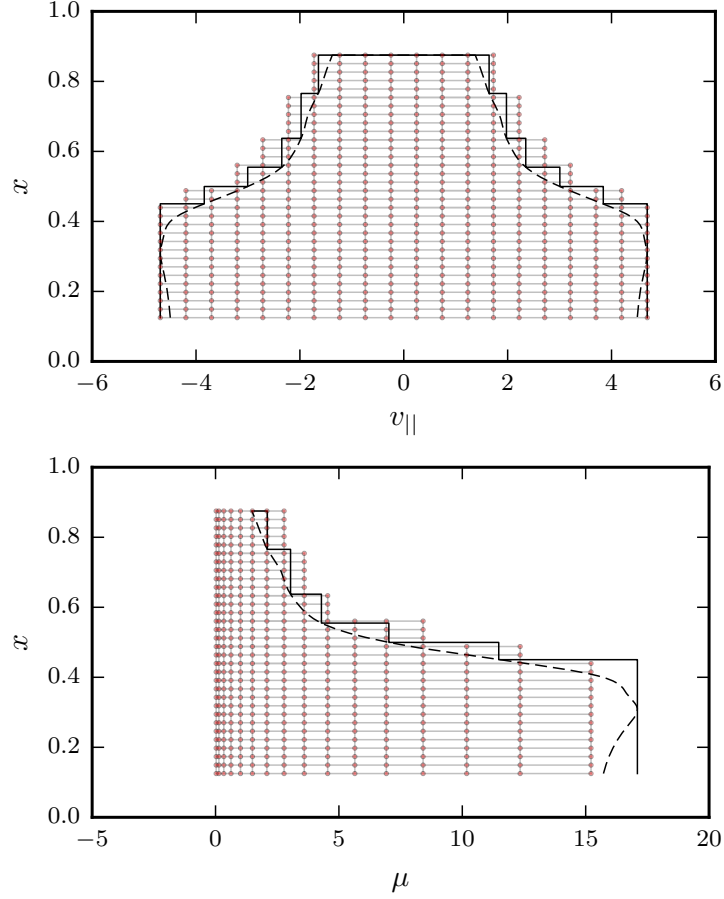


Figure 3: The first type of block-structured grids in $v_{||} - x$ and $\mu - x$ subspaces with confidence interval 99% and six blocks. The dash-lines show the contours of the simulation domain and the solid lines are approximations of these contours.

subspace, the achieved node reduction in this subspace is only around 23%. This happens because in the $\mu - x$ subspace the nodes are already located optimally for integrating exponential form functions in accordance to the corresponding distribution (10). As a result, most of the points are located close to the $\mu = 0$ axis, and neglecting a big area far from the axis does not lead to a significant reduction in the number of computational nodes.

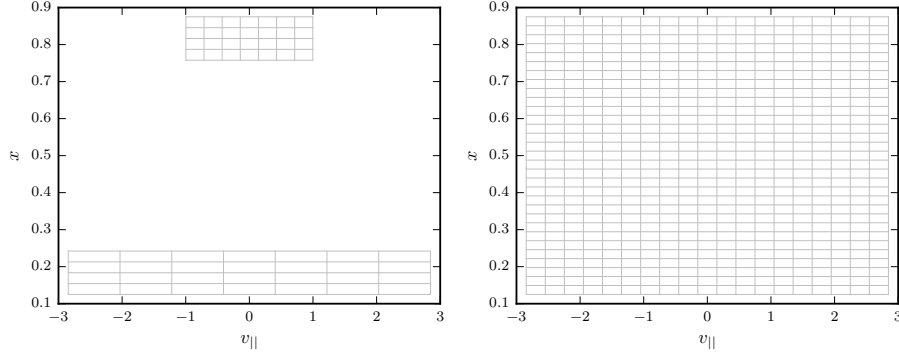
In the gyrokinetic simulations, the execution time of one time step is proportional to the number of grid points in the $v_{||}$ and μ directions. This means that the simulation run time (for the same number of time steps) in the previous example can be reduced to the theoretical value of 62%.

Table 2: Number of nodes in different grids for TCV profile simulations.

space type	grid type		
	regular	BS 1 6 blocks	BS 2 6 blocks
radial direction nx0	512	223+33+38+56+87+75	223+33+38+56+87+75
binormal direction nky0	1	1	1
parallel direction nz0	16	16	16
parallel velocity nv0	92	92,76,60,48,40,34	34
magnetic moment nw0	64	64,58,49,39,34,28	64,58,49,39,34,28
$v_{\parallel} - x$ subspace	47104	34022 (72%)	17408 (37%)
$\mu - x$ subspace	32768	25290 (77%)	25290 (77%)
full space	3014656	1864760 (62%)	859860 (29%)

To further reduce the number of points, we perform another improvement to yield our second type of block-structured grids, labeled “BS 2” in Table 2: in addition to adjusting the ranges of the parallel velocity direction v_{\parallel} to the temperature profile, we consider different v_{\parallel} resolutions.

The equilibrium distribution in the $v_{\parallel} - x$ subspace takes the form of a Gaussian bell curve. The actual shape of this curve depends on the temperature: for example, in high temperature regions the bell is wide with a rather flat peak, while in low temperature regions the bell is narrow with a sharp peak, as illustrated in Figure 2 (top). This fact is used to construct regular grids for local simulations, where the $v_{\parallel} - x$ subgrids have long v_{\parallel} ranges, but low resolutions in the high temperature regions, and short v_{\parallel} ranges, but high resolutions in the low temperature regions. A proper resolution for the local simulations is gained automatically by fixing the number of v_{\parallel} points and varying only the v_{\parallel} range according to the equilibrium distribution. The global regular grid is constructed by choosing the maximum v_{\parallel} range and resolution, as shown in Figure 4.

Figure 4: An example of two local grids (left) and resulting regular grid (right) in the $v_{\parallel} - x$ subspace.

There are less restrictions for the block-structured grids compared to the regular ones, because the resolution can be adjusted in each block. An example

of a second type of block-structured grids for the TCV profile is shown in Figure 5, where the fixed number of v_{\parallel} points in each block simplifies the domain decomposition for parallel computing (see Section 4).

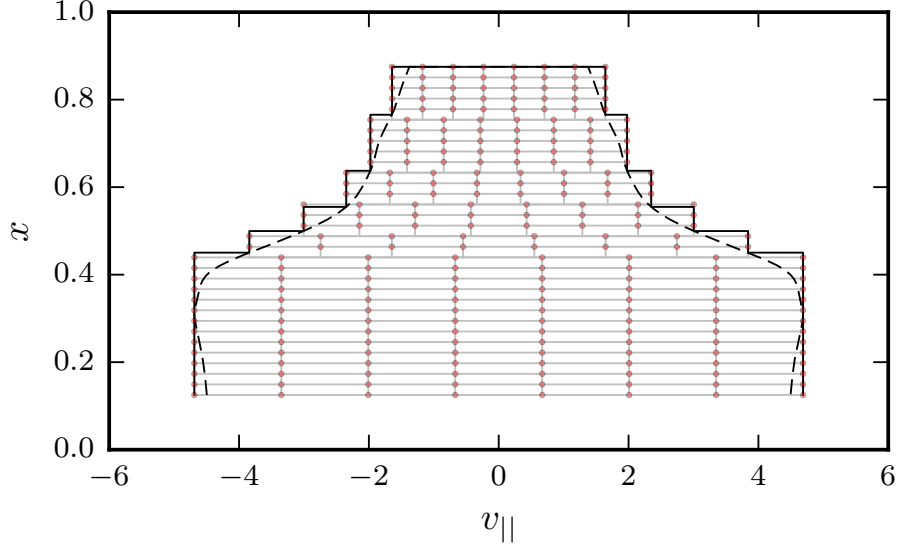


Figure 5: The second type of block-structured grids in $v_{\parallel} - x$ subspace with confidence interval 99% and six blocks. The dash-line shows the contour of the simulation domain and the solid line is an approximation of this contours.

Further insight into the rationale behind the construction of the block-structured grids of the second type can be obtained by looking at the parallel velocity coordinate scaled by the thermal velocity $v_{\parallel}/v_T(x)$, then, to obtain a well-resolved grid, we would have to keep the same resolution of this quantity at all radial positions. The block-structured grid of the first type (cf. Figure 3 (top)) has a good range in v_{\parallel} direction, but the resolution of $v_{\parallel}/v_T(x)$ is getting coarser in the upper blocks. Fixing the number of v_{\parallel} points in each block, the way it is done in the block-structured grid of the second type, yields approximately the same resolution of $v_{\parallel}/v_T(x)$ in all blocks.

This second type of constructing block-structured grids can also be applied to the $\mu - x$ subspace, which is part of our future work. It is technically more challenging, due to the gyrokinetic averaging and the more complex arrangement of grid points in the μ direction. Furthermore, when a non-uniform temperature profile is used, the thermal gyro-radius, which characterizes the scale length of the drift wave turbulence, can vary significantly. Due to these variations, the radial and binormal coordinates demand finer resolutions in the lower temperature regions. This problem can also be addressed by choosing the resolution of the x and y grids in each block of the block-structured grid. The adjustment of the discretization in the radial direction is also part of our future work.

The reduction in the number of grid points for the second type block-structured grid is demonstrated in Table 1 column “BS 2”. In this example, we removed more than 2/3 of all nodes.

Another important aspect in the construction of block-structured grids is choosing the v_{\parallel} and μ ranges ($1v$ and $1w$), structure and number of blocks.

When proper values for $1v$ and $1w$ are known from the regular grid simulations, we can assign these values to a reference point x_{ref} (usually located in the middle of the x range) to compute the confidence levels that correspond to the intervals $[-1v, 1v]$ and $[0, 1w]$ for the background distribution at x_{ref} . Then we use these confidence levels to find the contours of subdomains in the $v_{\parallel} - x$ and $\mu - x$ subspaces, see the dash-lines in Figures 3 and 5. By applying this procedure we usually obtain confidence levels more than 99%. If suitable values of $1v$ and $1w$ are not available, we directly choose confidence intervals around 99% as starting values to find the contours of the subdomains.

After we compute the contour of the domain, we approximate it by blocks. We ensure that the blocked approximation covers a larger area than the contour itself (see, for example, the solid lines in Figures 3 and 5). During the approximation procedure we minimize the difference between the probabilities of finding a plasma particle within the domain enclosed by the blocked approximation and the original contour.

To choose an optimal number of blocks in our grids, we look at how the number of grid points depends on the number of blocks. An example of such a dependence for the TCV profiles and the reference regular grid with $\text{nx0} = 512$, $\text{nv0} = 92$, and $\text{nw0} = 64$ is shown in Figure 6. The number of grid points in

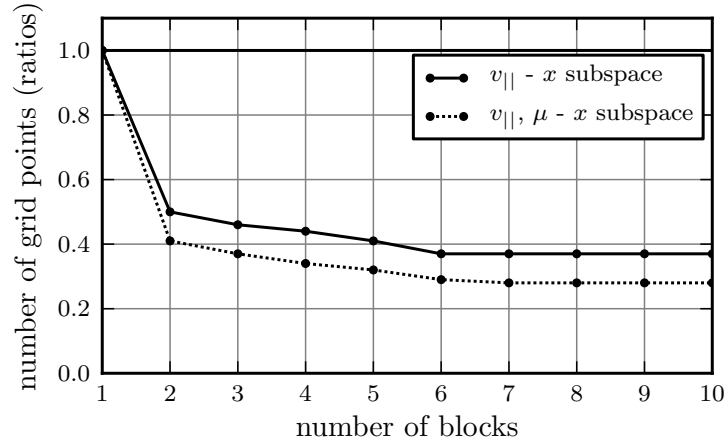


Figure 6: Dependence of the number of grid points on the number of blocks. The solid line corresponds to the second type block-structured grid in $v_{\parallel} - x$ subspace, the dot-line corresponds to the first type in $\mu - x$ subspace and second type in $v_{\parallel} - x$ subspace combined block-structured grid.

comparison to the reference grid, which corresponds to a block-structured grid

with one block, is indicated in ratios in the figure. In this example, the second type of block-structured grid was used for the $v_{\parallel} - x$ subspace and the first type for the $\mu - x$ subspace. From the plot, we see that the number of grid points no longer changes after six (seven) blocks. This means it does not make sense to take a higher number of blocks, because it does not reduce the total number of points in the grid. Furthermore, a higher number of blocks would lead to additional computational overhead on the block boundaries. For this particular example and with our current implementation, five to six blocks was the optimal choice.

The default finite difference scheme for the first order derivative computations in GENE is of the fourth order and provided in the “in” row in Table 3. This scheme can still be applied to the inner block nodes in the block-structured

Table 3: Fourth order finite difference scheme for first order derivatives. Node types: **in** inner node, **nb1** first boundary node, **nb2** second boundary node.

node type	derivative $f_x(x_i)\Delta x \approx \sum a_k f(x_k)$								accuracy
	a_{i-3}	a_{i-2}	a_{i-1}	a_i	a_{i+1}	a_{i+2}	a_{i+3}	a_{i+4}	
in		$\frac{1}{12}$	$-\frac{8}{12}$		$\frac{8}{12}$	$-\frac{1}{12}$			$\mathcal{O}(\Delta x^4)$
nb1		$\frac{1}{12}$	$-\frac{8}{12}$		$\frac{8}{12}$	$-\frac{1}{12}$			$\mathcal{O}(\Delta x^4) + \mathcal{O}\left(\frac{\Delta v_{\parallel}^m}{\Delta x}\right)$
nb2		$\frac{1}{12}$	$-\frac{8}{12}$		$\frac{8}{12}$	$-\frac{1}{12}$			

grids. However, due to the misalignment of the grid nodes in the second type block-structured grids (see Figure 5) the block boundaries have to be treated separately. This does not allow applying a finite difference scheme directly on the nodes close to the block boundary. The simplest way to solve this problem is to interpolate the distribution function at locations that are aligned with the adjacent block points, in order that derivatives can be easily computed on the boundaries of the neighboring block, see Figure 7. The stencil values that are multiplied with the interpolated values are shown in gray in Table 3. The application of the standard stencil on the interpolated values leads to errors of order $\mathcal{O}(\Delta v_{\parallel}^m / \Delta x)$ in the computations of the first order derivatives, where Δx and Δv_{\parallel} are the mesh sizes in the radial and parallel velocity directions. We can consider this inaccuracy as an $m - 1$ order error, where m is the order of the interpolation (the inaccuracy of the interpolation is $\mathcal{O}(v_{\parallel}^m)$) along the v_{\parallel} coordinate lines. For example, in the case of linear interpolation (second order), we get an error $\mathcal{O}(\Delta v_{\parallel}^2 / \Delta x)$, which corresponds to the first order approximation of the block boundary conditions. For the results provided in Section 5, we used the fourth order $m = 4$ polynomial interpolation scheme provided in Table 4.

4. Implementation of block-structured grids

One major requirement for new grids in GENE is the ability to easily port code — already written or under development — for the regular grids. For the presented block-structured grids, the simulation code inside grid blocks is

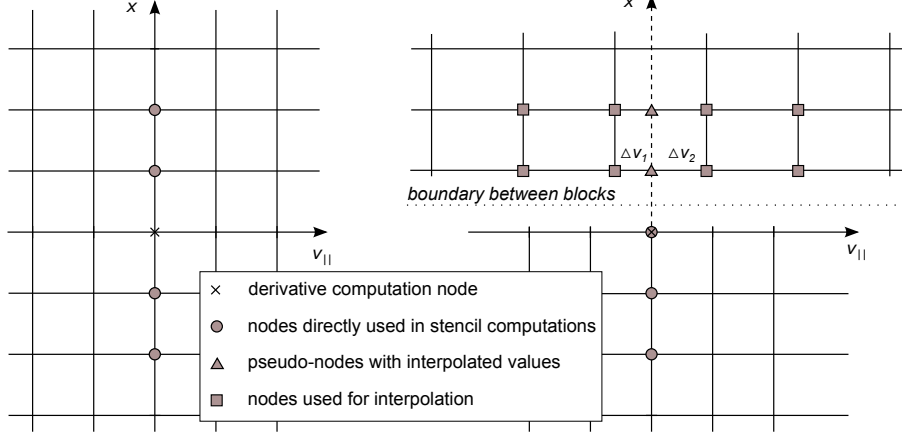


Figure 7: First order derivative finite difference stencil computations schemes for block inner nodes (left) and block boundary nodes (right).

Table 4: Fourth order polynomial interpolation coefficients for nodes outside the block.

function value $f(v_i + \Delta v_1)$ or $f(v_{i+1} - \Delta v_2)\Delta v_{ } \approx \sum b_k f(v_k)$ $\alpha = \Delta v_1 / \Delta v_{ }$ and $\beta = \Delta v_2 / \Delta v_{ }$			
b_{i-1}	b_i	b_{i+1}	b_{i+2}
$-\frac{1}{6}\alpha\beta(1+\beta)$	$\beta\left(1 + \frac{\alpha\beta}{2}\right)$	$\alpha\left(1 + \frac{\alpha\beta}{2}\right)$	$-\frac{1}{6}\alpha\beta(1+\alpha)$

the same as for the regular grids, modifications being required only on the boundaries of the blocks.

For the block-structured grids of the first type, we only have to consider the block boundaries that do not have neighboring nodes in adjacent blocks, and set the Dirichlet boundary condition for the distribution function to zero, see Figure 8 (left).

The situation changes for the block-structured grids of the second type. Now we have to interpolate in order to compute the first order radial derivatives close to the boundary. For this purpose, we introduce virtual or ghost grids in each block, which are prolongations of the neighboring block meshes. This is shown in Figure 8 (right). The values in a ghost block are interpolated as soon as a main block is updated. Consequently, when we have to compute the derivatives on the boundary, the interpolated values are already available.

The points of the ghost grids are not computationally equivalent to the points of the real grid, because the evolution of the distribution function is not computed at these points and they are used just to save the interpolated values. The ghost grids can only slightly increase the memory requirements of the block-structured grids, for instance, in the case of 6 blocks in Figure 6 the

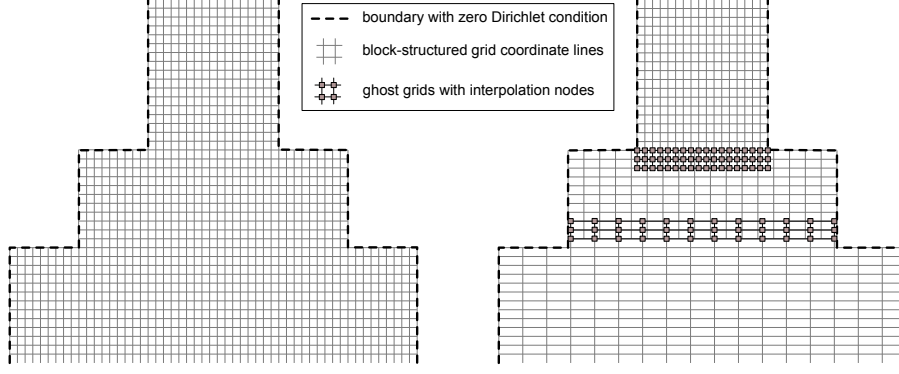


Figure 8: Fragments of the block-structured grids. Left figure: block-structured grid of the first type, right figure: of the second type, with ghost grids in the middle block.

number of points in the ghost grids is around 3% compared to the total number of the real grid points.

The domain decomposition in the $x - v_{\parallel}$ subspace for the distributed memory parallelization is simple for the regular grids. We decompose the domain into subdomains with approximately the same number of grid nodes, by dividing the x and v_{\parallel} ranges by a specified number of processes in these directions. An example with three subdivisions in both directions is shown in Figure 9 (top). The same approach is applied for the block-structured grids in the v_{\parallel} direction, where we subdivide the v_{\parallel} range of each block into equal parts. For the block-structured grids of the first type, however, we choose different locations for the grid boundaries of the processes in the x direction, to balance the number of grid points and, thus, of computational workload in each subdomain (see Figure 9 (bottom-left)). If such grids are also used in other directions, the number of grid nodes has to be balanced in the subspace spanned over these directions rather than on the plane, like in our example. This readjustment is not necessary for the block-structured grids of the second type, because in the latter case we preserve the number of points for each radial distance position. Therefore, the locations of the boundaries of the subdomains in the x direction is like that of the regular grids (see Figure 9 (bottom-right)).

From the previous examples, it is easy to see that exchange procedures in the v_{\parallel} direction are the same for all the discussed types of grids. Their implementation is standard: we first send to the left neighbor and receive from the right neighbor, and then send to the right and receive from the left. The situation changes for the exchange in the x direction. Now, beside the standard bottom-up or vertical exchange, we have to perform additional exchanges on the block boundaries, which are denoted by the side exchanges in Figure 9 (bottom). For the block-structured grids of the second type, the values sent on the block boundaries are not the actual values in the grid nodes, but the

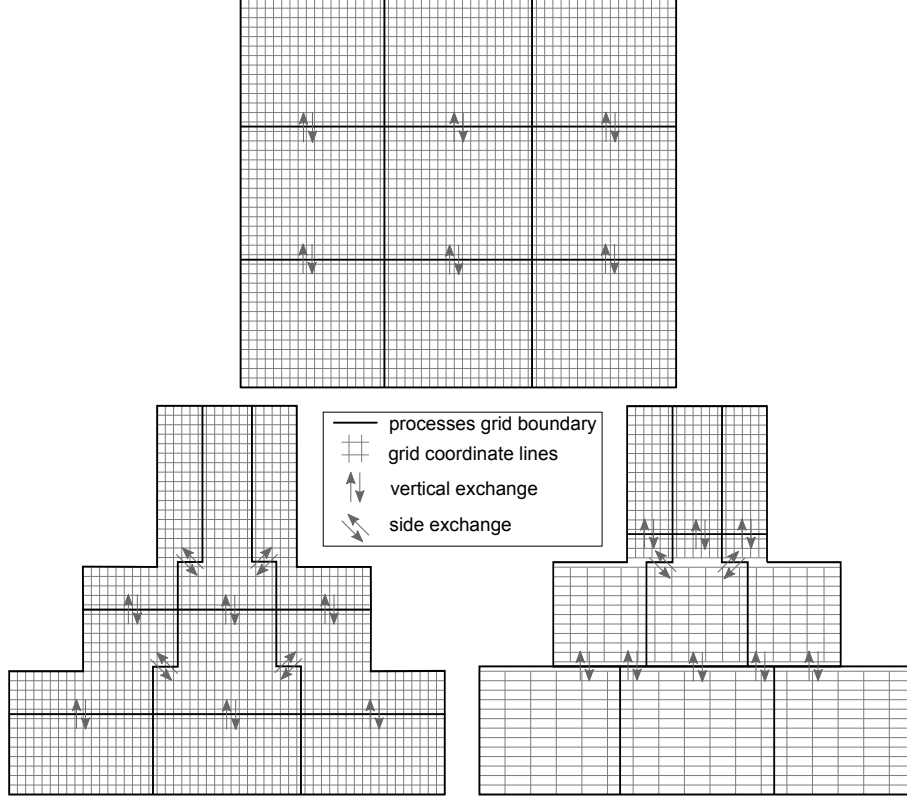


Figure 9: Examples of the domain decomposition in the $x - v_{\parallel}$ subspace for different grid types. Top figure: regular grid, bottom-left figure: first type block-structured grid, bottom-right figure: second type block-structured grid.

interpolated values from the ghost grids. If we do not have enough points in the process subdomain to interpolate some of the values, we send only the part of the interpolation sum that can be computed, while the other part is computed by the neighboring process. The receiving process has just to sum up these parts to get the correct interpolation value. The side exchanges can potentially be done by overlapping computations with communications once the main vertical exchange is done.

The complex implementation of the exchange and interpolation routines is hidden in a specially developed data-structure for block-structured grids. Each process has a copy of this data-structure, which holds not only distribution function values at each grid point in the subdomain, but also information about the local topology of the grid and the ghost grids, and synchronization information such as the processes involved in the communication and the order of exchange routines.

5. Results

While in the long term we focus on nonlinear gyrokinetic simulations (see Subsection 5.2), linear simulations still play an important role as a benchmark and are crucial for development, because they are computationally considerably less expensive than the equivalent nonlinear simulations. Furthermore, compared to nonlinear runs, linear simulations have clear stopping criteria: a linear simulation finishes when the measured linear growth rate and frequency of the dominant fluctuation mode have reached a certain prescribed precision.

5.1. Linear simulations

The linear instabilities can be calculated using either initial value simulations or eigenvalue computations. Details on the comparison of the two types are available in [32]. In this subsection, we provide results for the linear simulations implemented as an initial value problem.

To check the correctness and estimate the quality of the implementation of the block-structured grids, we compare the computed growth rate (γ) and frequency (ω) of the block-structured grids of the first and second type with values obtained from a regular fine-resolution reference grid.

Results of linear simulations for three different grids for the TCV electron profiles are demonstrated in Table 5, where **nv0** is the number of v_{\parallel} points, γ and ω are the growth rate and frequency of the dominant fluctuation mode, respectively, *steps* is the number of time steps until the results converge, Δt the cost in seconds of one time step, *time* the total simulation time, and *speedup* is the speedup in time of block-structured grids in comparison to the regular reference grid. The number of grid nodes in the other four directions besides v_{\parallel}

Table 5: TCV linear simulations results (growth rate, frequency, number of steps, time step cost, total simulation time, and speedup) of three types of grid: regular, first and second type of block-structured grids. Values of **nv0** in the “BS 1” column correspond to the number of grid points in the parallel velocity direction in each block for the first type of block-structured grids.

	grid type		
	regular	BS 1 5 blocks	BS 2 5 blocks
nv0	96	96, 78, 50, 40, 36	36
γ	0.371	0.371	0.357
ω	-0.056	-0.056	-0.050
<i>steps</i>	35359	35359	7779
Δt (s)	0.808	0.521	0.390
<i>time</i> (s)	28577	18415	3035
<i>speedup</i>	-	1.6	9.4

was fixed for all the three types of grids (**nx0**, **nky0**, **nz0**, **nw0**) = (128, 1, 16, 64) and the toroidal mode number was fixed to $k_y = 0.3$.

From the “regular” and “BS 1” columns in Table 5, we observe that the results (γ and ω) of the block-structured grid of the first type are identical with those of the fine regular grid for at least three digits after the decimal point. Furthermore, without losing precision, we gain a speedup of 1.6 in computational time. Similar observations were made for other linear simulations that we tested. This suggests that, for linear simulations, adjusting the v_{\parallel} range to the background distribution function, like in our block-structured grids, leads to results of the same precision as those of regular grids of the same resolution.

Adjusting not only the v_{\parallel} range, but also the resolution, yields significantly faster simulations, but with some accuracy penalty, which depends on the particular simulation scenario. In the provided example (see Table 5, column “BS 2”), we gain a total speedup of 9.4 for simulations with the block-structured grids of the second type, while the relative error value of the growth rate (γ) is small, around 3.8% only. Furthermore, it should be taken into account that the performance results provided in the table are given for the non-optimized block-structured grids. Performance tuning, which is currently under development, is expected to yield a cost of approximately 0.303 seconds for a time step and a total speedup of 12.

Next, it is important to make sure that by increasing the number of v_{\parallel} points ($nv0$) in the block-structured grid of the second type, we approach the results of the regular reference grid. Convergence curves of γ and ω for the block-structured grids of the second type and regular grids are shown in Figure 10. All results are shown in ratios (scaled by the values of the reference grid, which

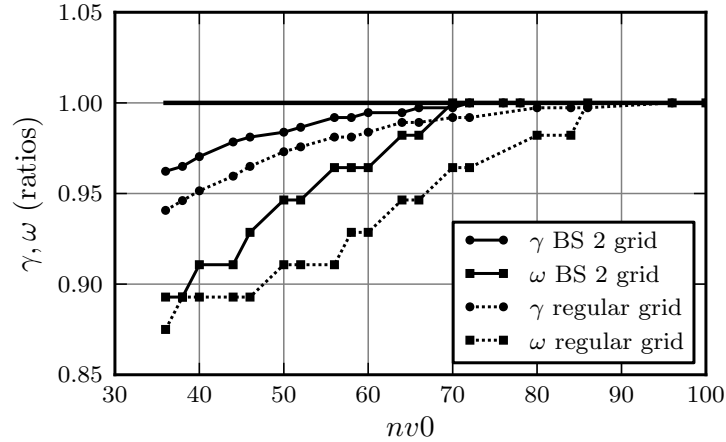


Figure 10: Convergence of γ and ω for the block structured grids of the second type for linear simulations with electrons TCV profiles. The solid lines correspond to the growth rate and frequency of the dominant fluctuation mode for simulations using second type of block-structured grids (“BS 2”) and the dot-lines correspond to simulations using regular grids.

has $nv0 = 96$ grid points, which corresponds to an accuracy of three decimal places), so that they can be easily associated with the relative error. The relative

error of the frequency (ω) in this case can look worse than it is, as its absolute value is quite close to zero, and frequencies can even cross zero. From the convergence plots, we observe that the block-structured grid of the second type yields more accurate results than a regular grid with the same number of v_{\parallel} points, and reaches the reference grid values with $\text{nv}0 \approx 70$ points.

To make a fair comparison between the block-structured grid of the second type and the regular grid, we have to compare the performance at the $\text{nv}0$ when the convergence for γ and ω is reached. From Figure 10 we observe that the results converge at $\text{nv}0 \approx 70$ for the second type block-structured grid and at $\text{nv}0 \approx 90$ for the regular grid. Furthermore, both grids take approximately the same number of time steps to converge at the specified $\text{nv}0$ values. The performance is therefore determined only by the time-step costs; the speedup is around $90/70 \approx 1.3$, which is less than the speedup of the block-structured grid of the first type (≈ 1.6). The reason for this is that at the end of the discussed linear run, only one dominant mode with a localized pattern is left, which does not comply with our assumption that the fluctuating part of the distribution function has a structure similar to the background distribution function. The fluctuating part of the distribution function on the $x - v_{\parallel}$ plane for this example is shown in Figure 11, where the absolute value of the distribution function is that in the middle of the parallel direction z and with a minimum magnetic moment μ . In this case, the accuracy of the simulation results depends only on

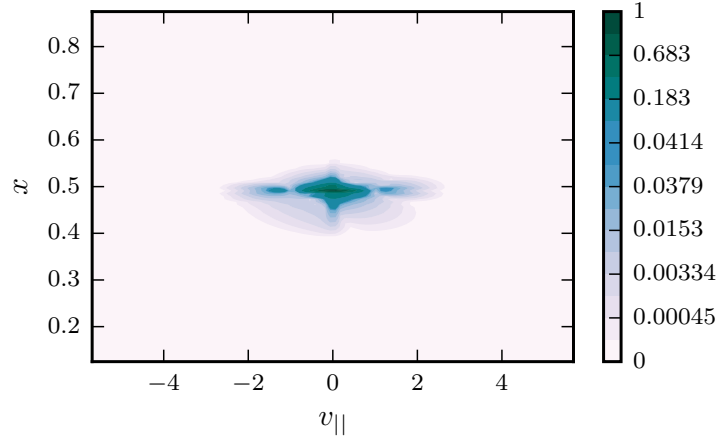


Figure 11: Fluctuating part of the distribution function absolute value on the $x - v_{\parallel}$ plane for TCv electrons. In the plot, we use adapted color-maps, to combine the benefits of both linear and logarithmic color-maps, without suffering from their individual disadvantages. They allow demonstrating the fluctuation near the zero value of the parallel velocity (similarly to linear color-maps), and show well the shape of the fluctuating part, without hiding the details close to the zero parallel velocity (like logarithmic color-maps do).

the resolution and range of the grid in the area, where the fluctuation part of the distribution function is localized. In this particular example, it is located

in the middle of the radial range and it fits into the second block of the block-structured grid. Therefore, the regular grid and the block-structured grid of the first type achieve the desired accuracy at a quite low nv0 , even though the lower temperature regions (in the upper part of the radial range) are not well resolved. The block-structured grid of the second type is designed to achieve a good resolution for the parallel velocity coordinate at all radial positions. As a result, in the particular example, it has more grid points than the first type of the block-structured grids. However, the block-structured grids of the second type are useful even for this type of linear simulations with a localized dominant mode. Before running the simulations, the localization pattern of the dominant mode is unknown. Computationally inexpensive simulations can be performed to find this pattern, by using the block-structured grid of the second type with a small number of grid points in the parallel velocity direction, which can achieve an acceptable precision. To improve these initial results, the simulation can then be repeated with a carefully chosen regular grid, where the radial distance, and parallel velocity ranges and resolutions have been appropriately adjusted.

The localized pattern of the dominating mode does not occur in all scenarios. An example of a fluctuating distribution function with a quite wide radial length is shown in Figure 12. In this example, we have a two species simulation, where

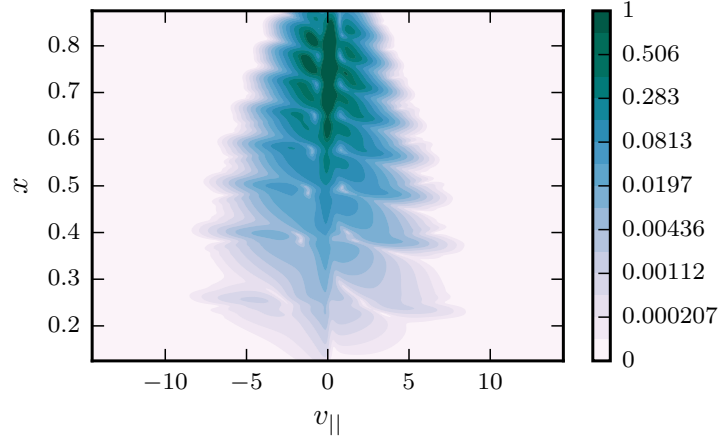


Figure 12: Absolute value of the fluctuating part of the distribution function on the $x - v_{||}$ plane for linear simulations with two species; the fluctuating part is shown for electrons only.

the regular grid resolution is $(\text{nx0}, \text{nky0}, \text{nz0}, \text{nv0}, \text{nw0}) = (256, 1, 16, 260, 64)$ and the toroidal wave number is $k_y = 0.3$. In Figure 12 the (μ, z) are fixed: the minimum of the magnetic moment and the middle of the parallel direction — outboard midplane position. To compare the block-structured grid of the second type with the regular grid, we provide convergence curves for γ and ω in Figure 13. In this example, the block-structured grid of the second type converges (i.e., accuracy three digits after comma) at $\text{nv0} \approx 100$ and the regular

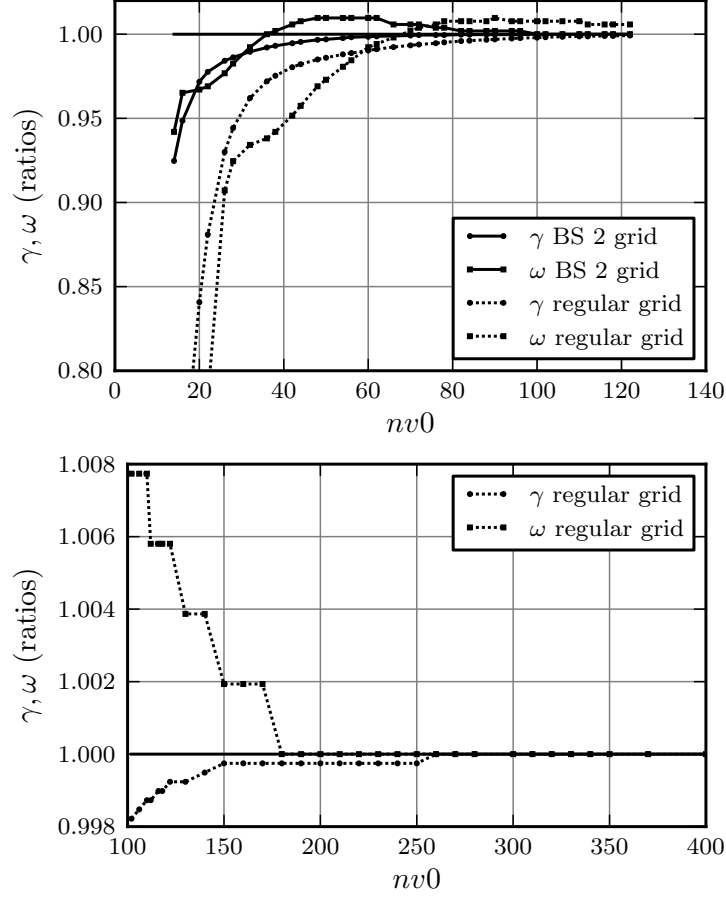


Figure 13: Convergence of γ and ω for the block structured grids of the second type, for the two species linear simulations.

grid at $nv0 \approx 260$. This result complies with the construction idea of the block-structured grid, as described in Subsection 3.2; the block-structured grid of the second type with $nv0 \approx 100$ is approximately equivalent to the regular grid with $nv0 \approx 260$.

The difference in the accuracy of the results between block-structured and regular grids with the same number of v_{\parallel} grid points becomes even more evident in the case of nonlinear results, because, in nonlinear global simulations, several modes can be excited simultaneously and fluctuations are observed almost everywhere in the selected $x - v_{\parallel}$ domain. Therefore, block-structured grids are especially beneficial for this type of simulations.

5.2. Nonlinear simulations

For the nonlinear simulations, we used the same TCV electrons profiles as for the linear runs. In this section, we provide results for the block-structured grids of the second type in the $x - v_{\parallel}$ subspace. As in case of the linear simulations, the block-structured grids of the first type yield accurate results for the nonlinear simulations, but the reduction of the grid points is not so high. Therefore, we decided to focus only on the second type of block-structured grids in future developments.

In nonlinear simulations, there are numerous observables that could be compared, for example, heat, particle, and momentum fluxes. However, these observables are usually obtained by integration in the five-dimensional space and sometimes hide important details. Therefore, we directly compare absolute values of the fluctuating part of the distribution function in the $x - v_{\parallel}$ subspace. Furthermore, due to the sensitivity to the initial conditions, the comparison of the nonlinear simulations results is less straightforward than for the linear simulations. Even small differences in computational grids can lead to significantly different distribution function values at the same time moment. For this reason, we compare the quasi-stationary features that are reflected in time-averaged values stemming from reasonably long time intervals, which do not include the initialization phase. An example of the time-averaged distribution function is shown in Figure 14.

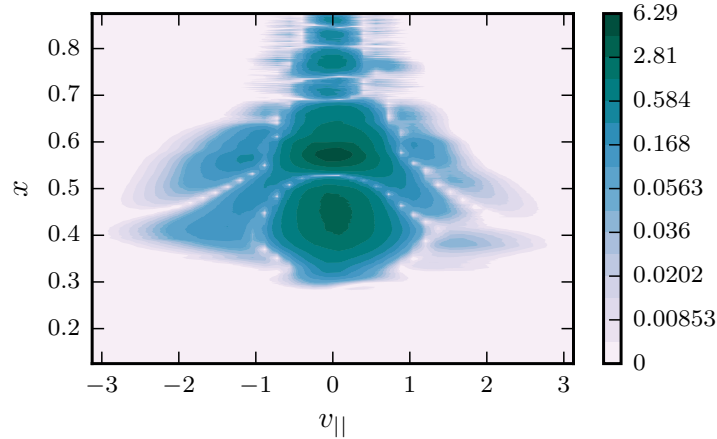


Figure 14: Time-averaged fluctuating part of the distribution function absolute value on the $x - v_{\parallel}$ plane for nonlinear simulations of TCV electrons.

In our block-structured grid tests, we compare results of simulations with a regular reference grid with the number of grid points $(nx0, nky0, nz0, nv0, nw0) = (512, 16, 16, 40, 64)$, and with two alternative regular grids, which have the same number of v_{\parallel} grid points as the block-structured grid ($nv0 = 18$). The reference grid has a wide range in the v_{\parallel} direction and is well-resolved in this

direction. The first alternative regular grid has a full v_{\parallel} range, but a coarse v_{\parallel} resolution. The second alternative regular grid has a reduced v_{\parallel} range, of the same dimension as the upper block of the block-structured grid, and a fine v_{\parallel} resolution. All these described grids are shown in Figure 15 (for the $x - v_{\parallel}$ projections). In this figure, however, for the sake of visibility, we coarsened the

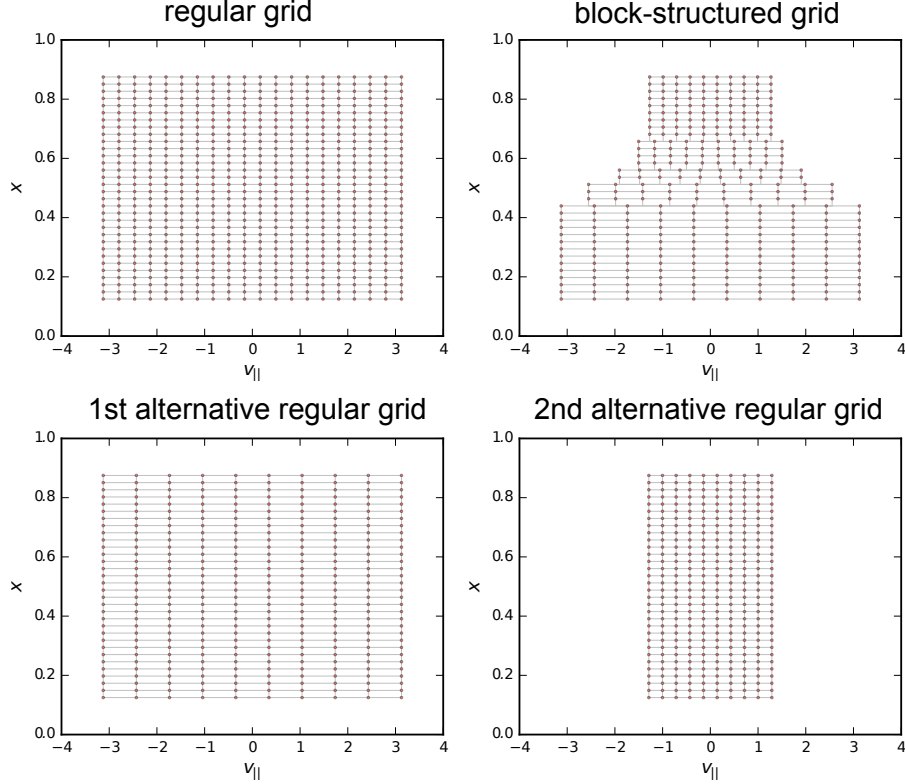


Figure 15: Examples of four $x - v_{\parallel}$ types of grids used to compare nonlinear results. Actual grids used in simulations have a finer resolution.

number of points in the x direction (vertical) by a factor of four and in the v_{\parallel} by a factor of two in comparison to the grids we used in our simulations.

The easiest way to compare the results stemming from all four grids is to produce plots over a line that goes through the maximum values of the time-averaged distribution function in the $x - v_{\parallel}$ subspace. The best choice is the line spanning the entire x range at a fixed $v_{\parallel} = 0$. In Figure 16, we provide the plots over lines, where the coordinates (k_y, μ, z) are fixed: k_y , the toroidal mode number, is zero, μ is the minimum value of magnetic moment, and z is the middle of the parallel direction interval (outboard midplane position). To produce Figures 14 and 16, we used the time-averaged distribution function on

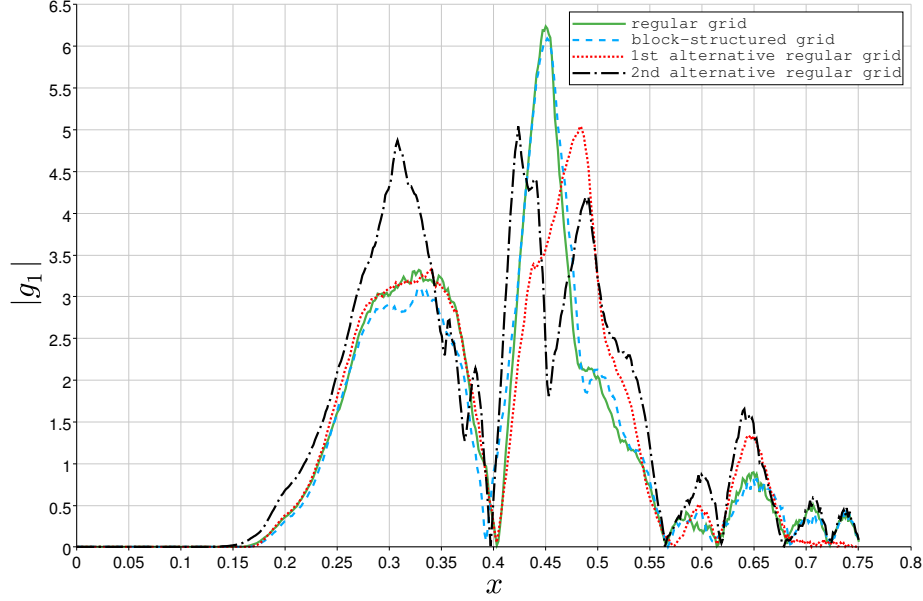


Figure 16: Plots over line (fluctuating part of the distribution function with fixed toroidal mode number $k_y = 0$) for four different computational grids. The solid line corresponds to the reference regular grid, the dash-line to the block-structured grid, the dot-line to the first alternative regular grid, with the wide v_{\parallel} range and coarse resolution, and the dash-dot-line to the second alternative regular grid, with the short v_{\parallel} range and fine resolution.

18 (R/c_s) time interval² with 30 – 40 samples at different time steps; for the comparison, the initialization phase of the simulations in this example takes around 6 (R/c_s). From the plots we observe that, while the results of the reference regular grid (solid line) match with the results of the block-structured grid (dash-line), the grid with the wide v_{\parallel} range and coarse resolution (dot-line) matches only in the low x values range (the first peak), and the grid with the short v_{\parallel} and fine resolution (dash-dot-line) gives correct results only for high x values range (the two last peaks). This demonstrates that the block-structured grid yields accurate results compared to the reference regular grids with less computational points.

Furthermore, we compare results for the dominant turbulence component $k_y \neq 0$, which drives the transport. To suppress the electron scale instabilities, a hyper-diffusion was added in the x and y directions. In this case, the regular reference grid had $(n_{x0}, n_{ky0}, n_{z0}, n_{v0}, n_{w0}) = (512, 16, 16, 82, 64)$ number of points, whereas the other three grids have $n_{v0} = 34$. These simulations yield the dominant mode with toroidal mode number $k_y = 0.6$. The heat fluxes averaged

² R — major radius, c_s — ion sound speed

over the time interval of 20 (R/c_s) with 40 – 50 samples at different time steps for all four grids are provided in Table 6. From the table, it can be observed that

Table 6: Heat fluxes in gyro-Bohm units averaged over a time interval of 20 (R/c_s) for four different computational grids.

	grid type			
	regular	block-structured	1st alternative	2nd alternative
Q_{gb}	1.63	1.62	1.53	3.70

only the block-structured grid gives results comparable to the reference regular grid. Additionally, the plots over a line (at fixed $v_{\parallel} = 0$) of the fluctuating part of the distribution function on the $x - v_{\parallel}$ plane are shown in Figure 17. The

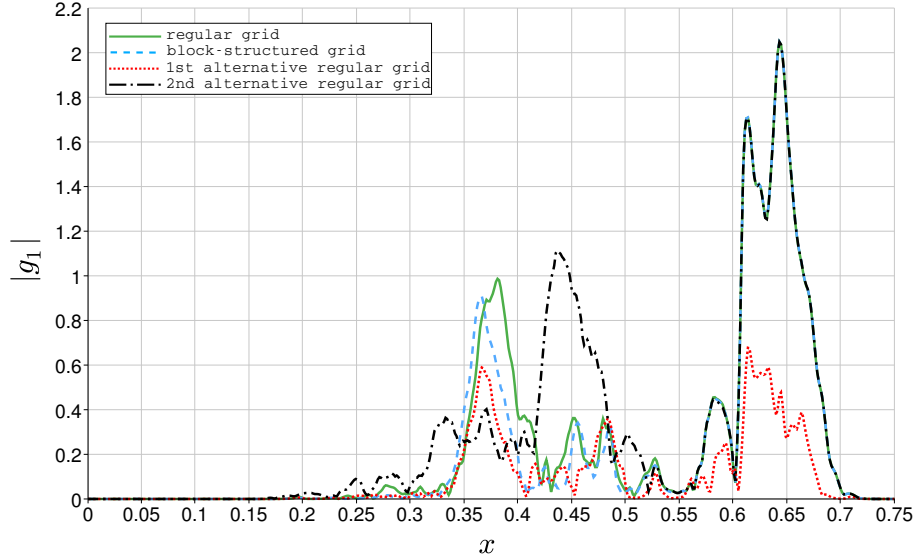


Figure 17: Plots over line (fluctuating part of the distribution function with fixed toroidal mode number $k_y = 0.6$) for four different computational grids.

blocked-structured grid (result shown in dash-line) has the closest match to the regular reference grid (result shown in solid-line).

The nonlinear simulations are significantly more expensive computationally than the linear runs. In order to be able to run physical scenarios in a reasonable amount of time using the available resources, it is extremely important to reduce the computational requirements. In the aforementioned example, the block-structured grid has a number of points that is approximately 2.2 times smaller than that of the corresponding reference regular grid. Theoretically, the speedup is also around 2.2. In practice, we found that the actual speedup depends on

the computer architecture and parallelization scheme chosen for the simulations (we used 32 - 64 CPUs for the presented nonlinear simulations), and can vary in the range 1.9 – 3.0.

For several other simulation scenarios, we observed that taking a very wide v_{\parallel} range for the high x values (low temperature range) can cause numerical instabilities. With the block-structured grids, the simulation can be stabilized by restricting this range in the low temperature region.

6. Conclusion

This paper details the application of block-structured grids in the context of gyrokinetic simulations. In comparison to other similar types of grids, e.g., consisting of multiple connected blocks to resolve complex geometries, the focus of the described block-structured grids is to adjust the mesh resolution and range of each block in the radial distance – velocity subspace. The exact shape of the domain boundary in the velocity subspace does not play an important role. Instead, the main factor determining the structure of the presented grids is the temperature radial profile, which we use to compute the background distribution function and then construct the block-structured grids.

Compared to regular grids that are often used in the velocity subspace of Eulerian gyrokinetic codes, block-structured grids possess several beneficial properties and may significantly reduce the computational requirements without losing accuracy. First of all, block-structured grids allow removing a substantial number of grid points while keeping good accuracy. Furthermore, the code for the inner blocks of these grids is equivalent to the already well-established and tested regular grid implementation; modifications are necessary only on the boundary.

We proposed two versions of block-structured grids. While the first version of these grids removes areas and corresponding grid points with the background distribution function close to zero, the second version additionally adjusts the parallel velocity coordinate resolution in each block. The first version of the block-structured grids served as an intermediate test case towards the second version, because the latter has a higher reduction in the number of grid points.

In the provided examples for the block-structured grids of the second type, we gained a maximum speedup of 9.4 for the linear runs and of 3.0 for the nonlinear runs. The reduced number of grid points resulted in a proportionally smaller one time-step execution, smaller memory footprint, and, for the linear simulations, smaller number of time steps necessary to reach convergence of the growth rate and frequency of the dominating mode. The memory footprint is especially important for the nonlinear runs, which require significantly more grid points in the radial direction. Despite the fact that the block-structured grids yielded smaller speedups for the nonlinear simulations than for the linear ones, the achieved improvement is more valuable, because nonlinear gyrokinetic simulations are extremely demanding in computational resources. Therefore, reducing the memory footprint and having three times faster simulations can

be decisive regarding what kind of simulations can be executed with the provided computational resources. More specifically, the fluctuating part of the distribution function is sensitive to the chosen parallel range and resolution in the nonlinear simulations. For this reason, the alternative regular grids with the same number of grid points as the block-structured grid were not adequate. On the contrary, the fluctuation shape from the simulations with the block-structured grid was quite close to the shape from the simulations with the regular fine-resolution reference grid.

The block-structured grids described in the paper do not worsen the parallelization properties of the original GENE code and should be considered as an additional improvement, rather than as an alternative to parallelization. The only additional side communications required are with neighboring processors in the v_{\parallel} direction, which are insignificant in comparison to the main data exchange. Furthermore, the load balancing for the second type of block-structured grids is as simple as for the regular grids.

Future work will address the extension of the second type of block-structured grids to the whole $v_{\parallel} - \mu$ subspace. Here, several challenges will have to be met. First of all, the gyro-averaging procedure or integration on the circular particles trajectories is more difficult when grid points are misaligned in the radial direction. Furthermore, the arrangement of grid points on boundaries of grid blocks in the $v_{\parallel} - \mu$ plane is more complex.

Acknowledgments

The support of HEPP (International Helmholtz Graduate School for Plasma Physics) is gratefully acknowledged.

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (FP7/2007-2013)/ERC Grant Agreement No. 277870.

- [1] A. J. Brizard, T. S. Hahm, Foundations of nonlinear gyrokinetic theory, *Reviews of Modern Physics* 79 (421) (2007) 421–468.
- [2] X. Garbet, Y. Idomura, L. Villard, T. Watanabe, Gyrokinetic simulations of turbulent transport, *Nuclear Fusion* 50 (4) (2010) 1–30.
- [3] International Thermonuclear Experimental Reactor, <https://www.iter.org>, accessed: 2015-04-07.
- [4] F. Jenko, W. Dorland, M. Kotschenreuther, B. Rogers, Electron temperature gradient driven turbulence, *Physics of Plasmas* 7 (5) (2000) 1904–1910.
- [5] T. Görler, X. Lapillonne, T. Brunner, T. Dannert, F. Jenko, F. Merz, D. Told, The global version of the gyrokinetic turbulence code GENE, *Journal of Computational Physics* 230 (18) (2011) 7053–7071.
- [6] Gyrokinetic Electromagnetic Numerical Experiment, <http://genecode.org>, accessed: 2015-03-03.

- [7] J. W. Connor, R. J. Hastie, J. B. Taylor, High mode number stability of an axisymmetric toroidal plasma, *Proceedings of the Royal Society A* 365 (1720) (1978) 1–17.
- [8] P. Xanthopoulos, F. Jenko, Clebsch-type coordinates for nonlinear gyrokinetics in generic toroidal configurations, *Physics of Plasmas* 13 (092301) (2006) 1–10.
- [9] F. Merz, Gyrokinetic simulations of multimode plasma turbulence, Ph.D. thesis, Mathematisch-Naturwissenschaftliche Fakultät der Westfälischen Wilhelms-Universität Münster (2008).
- [10] T. Görler, Multiscale effects in plasma microturbulence, Ph.D. thesis, Fakultät für Naturwissenschaften der Universität Ulm (2009).
- [11] X. Lapillonne, Local and global Eulerian gyrokinetic simulations of microturbulence in realistic geometry with applications to the TCV tokamak, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (2010).
- [12] M. A. Beer, Gyrofluid models of turbulent transport in tokamaks, Ph.D. thesis, Princeton University (1995).
- [13] J. Candy, R. E. Waltz, An Eulerian gyrokinetic-Maxwell solver, *Journal of Computational Physics* 186 (2) (2003) 545–581.
- [14] J. Candy, R. E. Waltz, Velocity-space resolution, entropy production, and upwind dissipation in Eulerian gyrokinetic simulations, *Physics of Plasmas* 13 (032310) (2006) 1–11.
- [15] T. H. Watanabe, H. Sugama, Velocity-space structures of distribution function in toroidal ion temperature gradient turbulence, *Nuclear Fusion* 46 (1) (2006) 24–32.
- [16] M. Barnes, W. Dorland, T. Tatsuno, Resolving velocity space dynamics in continuum gyrokinetics, *Physics of Plasmas* 17 (032106) (2010) 1–13.
- [17] J. Candy, C. Holland, R. E. Waltz, M. R. Fahey, B. E., Tokamak profile prediction using direct gyrokinetic and neoclassical simulation, *Physics of Plasmas* 16 (060704) (2009) 1–4.
- [18] M. Barnes, I. G. Abel, W. Dorland, T. Görler, G. W. Hammett, F. Jenko, Direct multiscale coupling of a transport code to gyrokinetic turbulence codes, *Physics of Plasmas* 17 (056109) (2010) 1–11.
- [19] J. Cary, R. Littlejohn, Noncanonical Hamiltonian mechanics and its application to magnetic field line flow, *Annals of Physics* 151 (1983) 1–34.
- [20] R. E. Denton, M. Kotschenreuther, δf Algorithm, *Journal of Computational Physics* 119 (2) (1995) 283–294.

- [21] Y. Chen, S. E. Parker, A δf particle method for gyrokinetic simulations with kinetic electrons and electromagnetic perturbations, *Journal of Computational Physics* 189 (2) (2003) 463–475.
- [22] J. Candy, R. E. Waltz, S. E. Parker, Y. Chen, Relevance of the parallel nonlinearity in gyrokinetic simulations of tokamak plasmas, *Physics of Plasmas* 13 (074501) (2006) 1–11.
- [23] W. D’haeseleer, W. Hitchon, J. Callen, J. Shohet, *Flux Coordinates and Magnetic Field Structure: A Guide to a Fundamental Tool of Plasma Theory*, Springer Verlag, New York, 1991.
- [24] J. A. F. Hittinger, J. W. Banks, Block-structured adaptive mesh refinement algorithms for Vlasov simulation, *Journal of Computational Physics* 241 (2013) 118–140.
- [25] J. L. Vay, P. Colella, A. Friedman, D. P. Grote, P. McCorquodale, D. B. Serafini, Implementations of mesh refinement schemes for Particle-In-Cell plasma simulations, *Computer Physics Communications* 164 (1–3) (2003) 297–305.
- [26] P. Colella, M. R. Dorr, D. D. Wake, Numerical solution of plasma fluid equations using locally refined grids, *Journal of Computational Physics* 152 (1999) 550–583.
- [27] G. Tóth, Y. Ma, T. I. Gombosi, Hall magnetohydrodynamics on block-adaptive grids, *Journal of Computational Physics* 227 (2008) 6967–6984.
- [28] H. Frerichs, D. Reiter, Y. Feng, D. Harting, Block-structured grids in Lagrangian 3D edge plasma transport simulations, *Computer Physics Communications* 181 (2010) 61–70.
- [29] G. Chesshire, W. D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, *Journal of Computational Physics* 90 (1990) 1–64.
- [30] S. E. Sherer, J. N. Scott, High-order compact finite-difference methods on general overset grids, *Journal of Computational Physics* 210 (2005) 459–496.
- [31] D. Told, Gyrokinetic microturbulence in transport barriers, Ph.D. thesis, Fakultät für Naturwissenschaften der Universität Ulm (2012).
- [32] M. Kotschenreuther, G. Rewoldt, W. M. Tang, Comparison of initial value and eigenvalue codes for kinetic toroidal plasma instabilities, *Computer Physics Communications* 88 (2–3) (1995) 128–140.