ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 185 (2003) 399–426

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Self-consistent modeling of turbulence and transport ☆

A.I. Shestakov, R.H. Cohen, J.A. Crotinger [1], L.L. LoDestro *, A. Tarditi [2], X.Q. Xu

Lawrence Livermore National Laboratory, Livermore, CA, USA

Received 13 February 2002; received in revised form 13 September 2002; accepted 6 November 2002

Abstract

We describe an efficient procedure for coupling a turbulent system to a transport equation which evolves the equilibrium fields that drive and are driven by the turbulence. As an example, we apply the procedure to the coupling of turbulence simulations of the two-dimensional Hasegawa–Wakatani equations to a one-dimensional transport equation for the density n. Our coupling scheme uses implicit temporal discretization of the transport equation, rendering it stable for arbitrarily large time steps. This allows the computation of turbulence with self-consistent steady-state equilibrium profiles in a single time step of the transport equations and with a total computational time comparable to that required for the turbulence code alone to reach a statistical steady state with fixed equilibrium profiles. Results are presented for both local and non-local turbulence simulations. In the former, which requires running a separate turbulence simulation for each transport grid cell, the transport flux $\Gamma(x)$ depends on only local values of $n(x)$ and $n'(x)$; for this case, Γ is expressed using Fick's law, $\Gamma = -Dn'$, with $D > 0$ prescribed by the turbulence. In the non-local simulations, $\Gamma(x)$ depends on the form of n over the entire domain. For such simulations we present two methods for representing Γ that allow for anomalous flux transport, i.e., regions where the flux flows up the local gradient of n.
© 2002 Elsevier Science B.V. All rights reserved.

AMS: 65M12; 76F25; 76F35; 76F65; 76R99; 65P10; 65P20; 77D05

1. Introduction

This paper describes a coupling scheme that connects two processes, each evolving on possibly vastly different time scales. The processes are microscopic turbulence in a fluid, driven by macroscopic inhomogeneities of the averaged fields, and the relatively slow evolution of the averaged fields under the influence of the turbulence.

☆ Work performed under the auspices of the US Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

* Corresponding author. Tel.: +925-422-6780; fax: +925-423-3484.

E-mail addresses: jcrotinger@mac.com (J.A. Crotinger), lodestro@llnl.gov (L.L. LoDestro), alfonso.g.tarditi1@jsc.nasa.gov (A. Tarditi).

[1] Present address: Proximation, LLC, 2960 Rodeo Park Dr. W., Santa Fe, NM 87505, USA.

[2] Present address: SAIC, 2200 Space Park Dr. Suite 200, Houston, TX 77058, USA.

0021-9991/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0021-9991(02)00063-3

Our motivating example is turbulent transport in toroidal magnetic-fusion-energy (MFE) plasmas, in which turbulent fluctuations in the plasma density, temperature, electrostatic potential, etc., are driven by gradients of the averages ("profiles") of these fields. Turbulence is considered to have a significant impact on the evolution of the averaged fields, and in particular to be the primary mechanism for transport of energy from the interior of the plasma to the surrounding structures. Quantitative modeling of this turbulence-transport system is thus essential to predict the performance of future large MFE machines. While considerable progress has been made in the understanding of the relevant turbulent processes [1], the derivation of analytic formulae for turbulent fluxes, valid for all relevant parameters and profiles, remains a formidable task. Another possible approach to incorporating turbulence processes in transport simulations is fitting databases of simulation [2] and/or experimental results. But this is also a daunting task if it is to span all relevant parameter regimes, particularly if the turbulent transport is non-local. In this paper we consider an alternative approach, which is direct coupling of turbulence and transport simulations, in a fashion which allows simulation for very long times, in particular, steady state.

There are two major challenges in undertaking a numerical coupling of turbulence and transport. One is how to resolve the two time scales efficiently. In a large tokamak, the transport evolves on a characteristic confinement time of the order of seconds, while the turbulent system is characterized by eddy turnover times of the order of a millisecond. The issue is complicated by the nature of the physical phenomena. Transport is often adequately modeled as a diffusive process (quantities usually flow from regions of higher to lower concentrations). On the other hand, turbulence is a rapidly changing, convective process. This difference has numerical implications: turbulent processes are usually simulated with explicit methods, since the details of the fluctuations must be followed. In contrast, for diffusive processes, which smooth out fast, short-scale features, implicit schemes are more efficient, since they allow the use of large time steps (large in comparison with the fastest normal mode). If times of the order of the transport time scale are to be simulated, a brute-force approach, whereby both processes are advanced at the same, smallest, time step, is prohibitively expensive. For relatively small tokamaks and for rather limited simulation times, such single-time step simulations have been carried out, either by solving for total (average plus fluctuating) quantities [3] or by solving separate equations for the fluctuating and average fields, e.g. [4].

The second challenge stems from the nature of the coupling: obtaining the transport flux Γ either wholly or in part from a turbulence simulation brings a set of difficulties. Each realization of Γ is fraught with noise. There is no a priori reason to assume that the fluxes are local, i.e., set by local values and derivatives of the profiles. In fact, there are experimental suggestions to the contrary [5]. The uncertainty of the analytic form of Γ can give rise to serious numerical difficulties. In particular, if Γ has regions of anomalous behavior, e.g., anti-diffusivity, then expressing it in terms of a Fick's law can lead to numerical instability.

We have addressed both of these issues via a scheme which separately advances the turbulence and transport equations on their own natural time scales, achieving implicitness in the transport equations through a relaxed iteration procedure. We outline our approach to the problem by considering a generic non-linear system,

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{\Gamma}(\mathbf{u}) = \mathbf{S}, \tag{1}$$

where $\mathbf{u}$ is the vector of dependent variables (density, temperature, etc.). We use a bar to denote averages (time, ensemble, or over some spatial dimensions), write $\mathbf{u} = \bar{\mathbf{u}} + \tilde{\mathbf{u}}$ where $\tilde{\bar{\mathbf{u}}} = 0$, and split the equations into their averaged and fluctuating parts:

$$\partial_t \bar{\mathbf{u}} + \nabla \cdot \bar{\mathbf{\Gamma}}(\mathbf{u}) = \bar{\mathbf{S}}, \tag{2}$$

$$\partial_t \tilde{\mathbf{u}} + \nabla \cdot [\mathbf{\Gamma}(\mathbf{u}) - \bar{\mathbf{\Gamma}}(\mathbf{u})] = \mathbf{S} - \bar{\mathbf{S}}. \tag{3}$$

Our method is predicated on the existence of disparate time scales τ and $\tilde{\tau}$ for Eqs. (2) and (3), respectively, and it will be most efficient when the solution of Eq. (2) is computationally much cheaper to obtain than that of Eq. (3), e.g., if it is of lower spatial dimension. However, even if the solution of Eq. (2) is of comparable expense to that of Eq. (3), there is still significant savings. If Δt and M define a time increment and the number of steps necessary to evolve Eq. (2) to its characteristic time, then $M\Delta t = \tau$. The analogous relationship for Eq. (3) is $\tilde{M}\Delta\tilde{t} = \tilde{\tau}$, where $\Delta\tilde{t}$ and $\tilde{M}$ are the time increment and the number of steps necessary to evolve Eq. (3) to its characteristic time, and $\tilde{\tau}$ is the time characteristic of the fluctuations in Eq. (3) to reach saturation, i.e., to approach a statistical steady state with fixed $\bar{\mathbf{u}}$. If one were to solve Eq. (1) directly, then the number of steps needed to study profile evolution $M_{\text{tot}} \sim \tau/\Delta\tilde{t} \sim \tilde{M} \times \tau/\tilde{\tau}$. This could be a very large number since $\tilde{M}$ is itself big. [3]

On the other hand, by separating Eq. (1) into two equations, we can solve each of them on its own characteristic time scale. Each time step of Eq. (2) entails a separate evolution of Eq. (3) until it saturates. Self-consistency is attained by iterating within each of the time increments of Eq. (2). At each iteration of Eq. (2), a turbulent time step is taken. For each iteration of Eq. (2), the latest $\tilde{\mathbf{u}}$ information from Eq. (3) is used in constructing $\bar{\mathbf{\Gamma}}$ and the latest $\bar{\mathbf{u}}$ from Eq. (2) is used where $\bar{\mathbf{u}}$ appears (e.g., the instability drives) in Eq. (3). Each transport time step Δt is iterated to convergence; the number of iterates is typically set by the requirement that the turbulence fluxes have saturated. Hence, each Δt time advancement yields a profile self-consistent with the flux at the advanced time; that is, our scheme is fully implicit. (Options to run with a mixture of advanced and old time step information are of course also available). Each Δt requires approximately $\tilde{M}$ iterative steps of Eq. (2) and $\tilde{M}$ time steps of Eq. (3). (Often, however, Eq. (2) is of lower dimension, so its advancement is relatively fast; in such cases the iterative overhead is negligible.). Thus we obtain $M_{\text{tot}} \sim \tilde{M} \times M$. As long as $M \ll \tau/\tilde{\tau}$, the method offers the possibility of a substantial improvement. As long as Eq. (2) is of lower dimension than Eq. (3), no additional large matrix solves have been introduced. Nor is there any need for Jacobian derivatives, which would be difficult to come by for this problem. Furthermore, our fully implicit differencing of Eq. (2) is unconditionally stable. The only limitations imposed on Δt are due to accuracy considerations. In particular, by letting $\Delta t \to \infty$ (so $M = 1$), we reach the saturated state in approximately $\tilde{M}$ steps.

The fully implicit discretization of Eq. (2) introduces complications. In this paper we restrict attention to the case where only one scalar quantity is evolved. We let l denote the iteration index for advancing Eq. (2) a single time step, with data exchanged with Eq. (3) each iteration as just described. Then each time increment of Eq. (2) requires iteratively solving

$$(\bar{u}^l - \bar{u}^o)/\Delta t + \partial_x \bar{\Gamma}^l = \bar{S}, \tag{4}$$

where $\bar{u}^o$ represents the solution at the old time level. Unfortunately, $\bar{\Gamma}$, a suitably averaged result from Eq. (3), depends on $\bar{u}^l$ implicitly (in the mathematical sense). To achieve numerical implicitness, we nevertheless give $\bar{\Gamma}$ a dependence on $\bar{u}$. One might do this by expressing $\bar{\Gamma}$ as diffusion, $\bar{\Gamma}^l = -D^{l-1}\partial_x \bar{u}^l$. However, in order to deal with the possibility of flux locally running up a gradient, we write it instead as a blend of a diffusive and a convective part. For a suitably chosen θ between 0 and 1, we write

$$\bar{\Gamma}^l = \theta(-D^{l-1}\partial_x \bar{u}^l) + (1-\theta)c^{l-1}\bar{u}^l.$$

In this expression, D and c, respectively, play the roles of a diffusion coefficient and a velocity. Both are constructed from the (suitably averaged) turbulent flux $\bar{\Gamma}$ and profile $\bar{u}$ (see Section 3). In Appendix A we present an alternative approach which, by adding an appropriate function to $\bar{u}$ (an alias), guarantees that $\bar{\Gamma}$ and the gradient of the dependent variable have opposite signs; then θ can be chosen to be one.

[3] Often there are important linear or non-linear frequencies in the problem much greater than $\tilde{\tau}^{-1}$. For fixed profiles, plasma turbulence codes typically need $\tilde{M}$ on the order of a few thousand steps to saturate, and more to achieve tight statistical requirements.

We conclude this section with a few remarks. The scheme is equivalent to a formal separation of scales, and so, apart from the usual errors associated with numerical derivatives, it introduces an additional error of order $\tilde{\tau}/\tau$ in the time-dependence of $\bar{\mathbf{u}}$.

If the iterations converge, the method is stable in time for arbitrary Δt. However, stability of the iterations to advance a single Δt is a serious concern. Analysis of simple one-field models reveals that the dependence of D on $\partial\bar{u}/\partial x$ introduces stability thresholds; stability for time steps large compared to a diffusive Courant limit requires averaging over prior iterates, or equivalently, relaxation. The relaxation required depends on the degree of non-linearity of $D(\partial\bar{u}/\partial x)$. A simple example is given in Section 2.

Given a stable iteration algorithm, the rate of convergence can also be an issue. Since in plasma turbulence problems the Eq. (3) solver typically takes a few hundred to a few thousand time steps to compute a reasonable average flux, there is not a very tight requirement on the convergence rate of the iteration algorithm; in particular, a large amount of relaxation in constructing D (to accommodate a highly non-linear D) can be tolerated. The coupling must not, however, significantly prolong saturation of the turbulence code, or the method loses its advantage. Feeding the turbulence code somewhat noisy $\bar{\mathbf{u}}$'s may also be a problem. In practice, we find that $\tilde{M}$ does not increase by more than a factor of 2–3, so that striking reductions in M_{tot} are achievable.

Finally, we remark briefly on code-coupling mechanics. A turbulence code whose dependent variable is $\mathbf{u}$ will of course have to be modified to solve Eq. (3) rather than Eq. (1). However, many low-amplitude turbulence codes are already written for fluctuating quantities that are in some sense small; e.g., they have already subtracted out "background" or "equilibrium" values. Such codes might retain in their calculations averages of "$\tilde{\mathbf{u}}$" that develop non-linearly. Such averages are, however, properly part of our $\bar{\mathbf{u}}$. If at $t = 0$, $\tilde{\bar{\mathbf{u}}} = 0$, then $\tilde{\bar{\mathbf{u}}} = 0$ for all time (since Eq. (3) averaged is $\partial\tilde{\bar{\mathbf{u}}}/\partial t = 0$). In order to employ our coupling algorithm, it is essential that a proper $\tilde{\mathbf{u}}$ be used, and that the turbulence code which evolves Eq. (3) allow $\bar{\mathbf{u}}$ (normally considered a fixed input quantity) to be changed every $\Delta\tilde{t}$ time step.

In the following sections we apply the above scheme to a simplified system consisting of a 1D transport equation for the background plasma electron density n, with the particle flux provided by a 2D simulation of plasma turbulence as modeled by the Hasegawa–Wakatani equations [6]. The processes are coupled by: (1) the flux of n, which is a statistical average of a saturated quantity of the turbulent system and (2) the background density n and its logarithmic gradient, which appear in coefficients of the turbulence equations; the gradient provides the source of free energy for the turbulence. We have successfully used two independent codes [7,8] to simulate the turbulent flux. These codes are explicit and therefore have stability conditions on their internal time steps. We first discuss transport–turbulence coupling in the *local* limit, where we use the local turbulence code HAWC [7]. Here the turbulence code is doubly periodic while the 1D transport domain has non-periodic boundary conditions. For local problems, each x grid point of the transport equation corresponds to a separate 2D turbulent domain. The physical assumption is that the turbulence depends on only the local average of $n(x)$ and $\mathrm{d}n/\mathrm{d}x$. In the regimes tested, the space- and time-*averaged* D is always positive, although instantaneous space-averaged values may go negative. Section 2 describes the methods and numerical results applicable to coupling to this local turbulence model. A model problem with an exact analytic solution is presented in Section 2.1, and in Section 2.2 we couple the transport equation to HAWC. Section 3 is devoted to the more difficult problem, coupling to *non-local* turbulence, for which $\Gamma(x)$ depends on the global form of $n(x)$. Here the possibility arises of non-local effects resulting in a flux whose analytic structure cannot be cast in the form $-D\,\mathrm{d}n/\mathrm{d}x$ with $D(n(x), n'(x), \ldots, x) > 0$. Indeed, we present results of problems which violate the eikonal assumption used to derive our turbulence equations in order to obtain locally anti-diffusive behavior and so test our coupling method in this regime. Section 3 describes our splitting of $\bar{\Gamma}$ into diffusive and convective parts. An alternative approach, the addition of an "adaptive alias" density, is described in Appendix A. Concluding remarks and a discussion of future work is presented in Section 4.

Alternative approaches to multiscale problems, which attempt a treatment in considerable generality, can be found in [9] and references therein. The applicability of the methods to plasma turbulence, where noise presents a serious complication, is not clear.

2. Problem description and local coupling

In this section, we first discuss some general features of our scalar transport equation and its coupling to a turbulence model. We next present results for which the turbulent flux is given by an analytic function. Then, in Section 2.2 we introduce the Hasegawa–Wakatani model turbulence equations, and present results coupling them, within the local approximation, to the transport equation. Henceforth, we drop the overbar symbol on space- (or ensemble-) averaged quantities and reserve its use to denote an average over iterations; fluctuating quantities will be noted explicitly where needed.

We are interested in solving the equation

$$\partial_t n + \partial_x \Gamma = S - \gamma n \quad (S, \gamma \geqslant 0), \tag{5}$$

$$\text{for } 0 \leqslant x \leqslant L_x, \tag{6}$$

where S and γ are, respectively, the source and absorption coefficients. The initial condition is $n|_{t=0} = n^0(x)$; and the boundary conditions are

$$a_i n + b_i \partial_v n = c_i \quad (i = 0, 1), \tag{7}$$

where ∂_v denotes the outward normal derivative. Standard conditions such as Dirichlet (fixed n) or Neumann (fixed flux) are obtained with appropriate choices for a_i, b_i, c_i.

The flux Γ is generally obtained numerically from another calculation, viz., the space, time, or ensemble average, $\tilde{\Gamma} = \langle \tilde{n} v_x \rangle$, of the solution of the turbulence equations. The flux $\tilde{\Gamma}$ is the described $\bar{\Gamma}$ of Section 1. Although it is an averaged quantity, in practice it is often quite noisy. This flux is used to define a diffusion coefficient (cf. Fick's law):

$$\tilde{D} \equiv -\tilde{\Gamma}/\partial_x n.$$

The flux Γ which is used in Eq. (5) is then written as $\Gamma = -\bar{D}\partial_x n$, where $\bar{D}$ is obtained from $\tilde{D}$ by averaging over previous iterates. (Alternatively, one can use $\bar{D} = -\bar{\Gamma}/\overline{\partial_x n}$; see discussion of averaging schemes below.) As discussed later in more detail, our linear transport-equation solver requires $D > 0$ and guarantees $n > 0$. In this section, we restrict attention to problems where the flux flows down local gradients so that indeed $D > 0$. The more general, possibly anti-diffusive case is deferred to Section 3.

The functional dependence of D on n or its derivatives is usually unknown. We now show that a dependence of D on the derivative of n requires a fully implicit time-differencing scheme to advance Eq. (5). Using m to denote the time level, consider the scheme

$$(n^m - n^{m-1})/\Delta t - \partial_x (D^i \partial_x n^m) = S - \gamma n^m \quad (\text{where } i = m \text{ or } i = m-1). \tag{8}$$

For simplicity, let $\gamma = 0$. The semi-implicit scheme ($i = m - 1$) can be unstable. To see this, let $n^m = n^e + \hat{n}^m$, where n^e solves: $-\partial_x (D^e \partial_x n^e) = S$. Assume $D = D_0 (\partial_x n)^p n^q$, linearize about n^e, and let $\hat{n}^m = \lambda^m e^{i\xi x}$. The instability condition $|\lambda|^2 > 1$ is:

$$2(a-c)D\Delta t + (D\Delta t)^2 \left(a^2 + b^2 - c^2 - d^2\right) > 0, \tag{9}$$

where

$$a = q(p+1)\frac{n_{xx}}{n} + q(q-1)\left(\frac{n_x}{n}\right)^2 - p\xi^2,$$

$$b = \left[p^2\frac{n_{xx}}{n_x} + q(p+1)\frac{n_x}{n}\right]\xi,$$

$$c = \xi^2,$$

$$d = -\left(p\frac{n_{xx}}{n_x} + q\frac{n_x}{n}\right)\xi.$$

Here subscripts denote partial derivatives and superscript "e" on D and n has been dropped.

For short-wavelength perturbations, $\xi \gg (n_x/n)$, (n_{xx}/n_x), and positive Δt, this criterion reduces to $(p+1)[-2+(p-1)\xi^2 D\Delta t] > 0$. This criterion is satisfied (instability) for $p < -1$ and any Δt, and also for $p > 1$ so long as $\xi^2 D\Delta t > 2/(p-1)$. For $|p| < 1$ we have short-wavelength, long-time step stability (but not necessarily stability for shorter time steps and longer wavelengths), and for $|p| = 1$ the answer depends on the sign of n_{xx}/n_x and the sizes of n_{xx}/n_x, n_x/n, and q.

For the particular case of constant diffusion ($p = q = 0$), we have $a = b = d = 0$, $c = \xi^2$, and the instability criterion becomes $-2\chi - \chi^2 > 0$, where $\chi \equiv \xi^2 D\Delta t$; this is not satisfied for any ξ or any positive Δt and so the scheme is stable.

On the other hand, for the Hasegawa–Wakatani equations, a scaling analysis (see [10]) indicates that for large values of the parameter α (defined in Section 2.2 below, just after Eq. (19)), $p = -q = 2$, whereas for small α, $p = -q = 4/3$, and numerical scans indicate a smooth dependence in between. Hence, for all α we have large-ξ instability for $\xi^2 D\Delta t \gg 1$.

On the other hand, if the scheme is fully implicit [$i = m$ in Eq. (8)], then Δt is limited only by accuracy considerations. This result was dramatically exhibited in one of our test problems. We used $p = -q = 2$, started with a constant initial state, an initial $\Delta t = 10^{-12}$, and let n relax to an equilibrium. The code increased Δt by a factor of 2 if n varied less than a factor of 2 from cycle to cycle. Using Eq. (8), and $i = m - 1$, after 513 time cycles, t had grown to 0.00234 but a time history of Δt showed increases followed by sharp drops indicating sudden changes in n. A closer inspection of n showed oscillatory behavior with a wavelength equal to twice the mesh width, a sign of numerical instability. However, when the same problem was run fully implicitly ($i = m$), after 34 cycles $\Delta t = 0.0170 = 2^{34} \times 10^{-12}$; i.e., despite the significantly larger steps, Δt grew by the maximum allotted amount with no deleterious behavior of the solution. Henceforth, in all our discussions we use $i = m$. Of course, this requires that Eq. (8) be iterated at each time level.

We now describe the scheme for coupling Eq. (5) to a turbulence simulation which computes Γ. We are interested in the case $\tau \gg \tilde{\tau}$, where τ is a characteristic time for Eq. (5) and $\tilde{\tau}$ is the non-linear saturation time for the turbulent flux. The respective equations are advanced with different time steps. Equation (5) is evolved in steps of size Δt and the turbulence code with a significantly smaller $\Delta\tilde{t}$. For each transport time step, the turbulence code is advanced many $\Delta\tilde{t}$ time steps until a saturated statistical steady state is reached. The turbulent time steps are iterative increments for the transport time step. Introducing $l \geqslant 1$ as the iteration index, Eq. (5) is repeatedly advanced using Eq. (8), with

$$n^m = n^{m,l} \quad \text{and} \quad D^i = \bar{D}^{m,l-1}, \tag{10}$$

where we remind the reader that m is the transport time step index. That is, we iterate on the time-advanced density n^m and evaluate the diffusion coefficient using the previous (time-advanced) iterate. The ensemble-averaged quantities ("profiles") appearing in the turbulence equations are not evolved by the turbulence

code, but rather, are set at each $\Delta\tilde{t}$ from the latest iterate of the transport equations (or from an average over recent iterates).

We consider various possibilities for computing $\bar{D}^{m,l-1}$ from the turbulent flux data. Each time cycle $\Delta\tilde{t}$ the turbulence code returns a value for the diffusion coefficient $\tilde{D}^{m,l-1}$. The $\tilde{D}$ coefficients are used to construct an averaged value. Two averaging schemes were investigated in some detail and are reported on below. These schemes are just representative possibilities; the important point is that in general some scheme to stabilize the iterations is necessary.

One averaging scheme smoothly relaxes D to the average of the iterates. First we choose a large number N ($=100$ by default) and define

$$\bar{D}^{m,l} = \begin{cases} \max(0, (1/N)\sum_{j=1}^{l} \tilde{D}^{m,j} + ((N-l)/N)\bar{D}^{m-1}) & \text{if } l < N, \\ \max(0, (1/l)\sum_{j=1}^{l} \tilde{D}^{m,j}) & \text{if } l \geqslant N, \end{cases} \tag{11}$$

where $\bar{D}^{m-1}$ is the converged D from the previous transport step. Since Eq. (8) is solved by iteration, we do not check for convergence until $l \geqslant N$. For such l's, Eq. (11) gives equal weight to all of the $\tilde{D}^{m,l}$.

Since the transport iterates correspond to turbulence-code time increments $\Delta\tilde{t}$, the early $\tilde{D}^{m,l}$ (low l), which are statistical fluctuations about some transient, may differ significantly from the $\tilde{D}^{m,l}$ with large l. The later ones are fluctuations about the desired mean $\bar{D}^{m}$. Thus, it may be better to weigh them more heavily. To this end, we also use the alternate scheme: for some large N with $l \geqslant N$ and $a = 1 - 1/N$, set

$$\bar{D}^{m,l} = \max\left(0, A_l \sum_{j=0}^{l} a^{l-j}\tilde{D}^{m,j}\right), \quad A_l = (1-a)/(1-a^{l+1}). \tag{12}$$

It is easy to show that Eq. (12) is a form of relaxation,

$$\bar{D}^{m,l} = A_l\tilde{D}^{m,l} + (1-A_l)\bar{D}^{m,l-1}.$$

Furthermore, $\lim_{l\to\infty} A_l = 1 - a$. For $l < N$, the iterations are typically begun as in Eqs. (11).

The local coupling results presented in Section 2.2 were obtained using Eqs. (11). The non-local coupling results in Section 3 were obtained using Eq. (12). We have also successfully used Eqs. (11) for non-local problems, but, as expected, with somewhat slower convergence. We perform a large-wavenumber stability analysis of either scheme as follows. Assume D is of the form given following Eq. (8), and denote with overbar$^{(l)}$ any averaging operator linear in the iterates j, $1 \leqslant j \leqslant l$ (thus, we consider $l \geqslant N$ in the schemes above). We perturb about a stationary (to iterations) solution of Eq. (8) and write $n^{m,l} = n^m + \delta n^{(l)}$. Then, for large perturbation wavenumbers, $\partial\bar{D}^{m,l}/\partial x = -k^2(\partial D^m/\partial n')\overline{\delta n}^{(l)}$, where $n' = \partial n^m/\partial x$ and k is the perturbation wavenumber. Then the perturbation of Eq. (8) becomes

$$p\overline{\delta n}^{(l)} + \delta n^{(l+1)} = 0, \tag{13}$$

where p was introduced just below Eq. (8). To analyze the iteration scheme of Eq. (11), subtract $(l-1)/l$ times Eq. (13) with $l-1$ substituted for l from Eq. (13) as it appears. Then, for $l > N$, upon using Eq. (11), we find $\delta n^{(l+1)} = \delta n^{(l)}[1 - (1/l) - (p/l)]$, which indicates convergence for $p > -1$, though the convergence becomes very slow for large l. For the scheme given by Eq. (12), we seek a solution for $\delta n^{(l)}$ of the form $\delta n_0\lambda^l$. Then $\overline{\delta n}^{(l)} = A_l\delta n_0(\lambda^{l+1} - a^{l+1})/(\lambda - a)$. If we pessimistically assume that the scheme is weakly convergent or non-convergent so that $(|\lambda|/a)^l \gg 1$ for large l, then in that limit $\overline{\delta n}^{(l)} \approx \delta n_0\lambda^{l+1}$ $(1-a)/(\lambda - a)$, and Eq. (13) yields $\lambda \approx 1 - (1+p)/N$. Thus we have stability provided that $p > -1$ and $N \geqslant \min(1, (1+p)/2)$. (However, one should be cautious in applying this result to the actual coupled code algorithms described in Sections 2 and 3 since the above analysis effectively assumes a converged turbulence calculation, so that D is uniquely a function of the present profile and not of the iteration history, whereas

the coupling schemes employed simultaneously iterate the transport equation and advance the turbulence equations.)

Finally, we remark that having defined these two averaging schemes, one could apply them instead to the flux and to the profiles, and then compute an average diffusion coefficient according to $\bar{D}^{m,l} = -\bar{\Gamma}^{m,l}/\overline{\partial_x n}^{m,l}$. Since the division is now by an averaged quantity, $\bar{D}$ is less subject to large changing iterates occasioned by $\partial_x n^{m,l}$ passing through zero. For cases with gradients that are either small or change sign in x, this can be important. The stability analyses given in the preceding paragraph apply without alteration to this modified definition of the average diffusion coefficient.

Now we turn attention to the spatial discretization. The turbulence models are treated as a "black box". For Eq. (5), the spatial discretization is done using finite elements. The piecewise-linear test functions satisfy $\phi_i(x_j) = \delta_{ij}$; hence, n has the representation

$$n^m(x) = \sum n_j^m \phi_j(x). \tag{14}$$

Eq. (8) is multiplied by a test function and integrated over the domain. The transport term is integrated by parts and the undifferentiated terms are lumped. This leads to a linear system for the unknowns n_j^m,

$$Mn^m = y.$$

If n^{m-1} and S_j are both non-negative, so is y. By construction, M is a symmetric M-matrix, which assures that with these conditions and appropriate boundary conditions n^m is non-negative [11].

The scheme is conservative. This allows us to compute (and account for) quantities such as the initial and final masses, the total mass flux, the total source, and the total mass absorbed, i.e., discrete analogues of

$$N^0 = \int \mathrm{d}x\, n^0(x), \quad N(t) = \int \mathrm{d}x\, n(x,t), \quad G(t) = \int_0^t \mathrm{d}t'\, \Gamma|_{x=0}^{L_x},$$
$$\Sigma(t) = \int_0^t \mathrm{d}t' \int \mathrm{d}x\, S, \quad \text{and} \quad B(t) = \int_0^t \mathrm{d}t' \int \mathrm{d}x\, \gamma n.$$

At the end of any time step, to machine accuracy,

$$N(t) + B(t) = N^0 - G(t) + \Sigma(t).$$

We now discuss one reason for the max function in Eqs. (11) and (12). Because the applied iteration-averaging described above does not completely eliminate noise in the diffusion coefficient $\tilde{D}^{m,l-1}$, $\bar{D}^{m,l-1}$ might turn out to be negative. Introducing the max function in combination with lumping guarantees the M-matrix property for each of the M's used in the iterative steps.

We caution the reader that it is easy to generate a singular matrix M if $\Gamma = -D\nabla n$ and D is allowed to be negative. The $\bar{\Gamma} \cdot \nabla n < 0$ condition is also necessary to guarantee certain expected properties of transport equations for positive quantities, viz., that maxima (minima) decrease (increase), that $n \geqslant 0$, etc. To see this, consider the equation $\partial_t n = -\partial_x \Gamma$ where $\Gamma \partial_x n > 0$. Assume that $n(x) = n(-x)$, $n(0) = \partial_x n(0) = 0$, and $\partial_{xx} n(0) > 0$. A Taylor series expansion shows that

$$n(0, \Delta t) = -\Delta t\, \partial_x \Gamma|_{t=0,x=0} + \mathrm{O}(\Delta t^2) = -(\Delta t/\Delta x)(\Gamma_{1/2} - \Gamma_{-1/2}) + \mathrm{O}(\Delta x^2 \Delta t) + \mathrm{O}(\Delta t^2) < 0,$$

where the inequality is a result of $\Gamma_{\pm 1/2} \cdot \partial_x n > 0$. The local minimum decreased causing a negative density.

Nevertheless, we do anticipate scenarios in which the flux displays locally anomalous behavior, i.e., regions where $\Gamma \cdot \nabla n > 0$, particularly if the flux is responding to, say, a spatially averaged (x as well as y), rather than a local gradient. This is the subject of Section 3.

2.1. Analytic test problem

We present results for an analytically tractable problem, the computation of the steady-state solution of Eq. (5) in which we set

$$\gamma = 0, \quad \Gamma = -l_n^{-2}\partial_x n, \quad l_n = |n/\partial_x n|. \tag{15}$$

In Eq. (6) we set $L_x = 1$. The boundary conditions, Eqs. (7), are imposed by setting $a_0 = c_0 = b_1 = c_1 = 0$ and $b_0 = a_1 = 1$. The initial condition is $n^0(x) = 1$. The source is a step function,

$$S(x) = \begin{cases} S_0 & x < \delta, \\ 0 & \text{otherwise}, \end{cases} \qquad S_0 = 1, \ \ \delta = 0.1.$$

This models the decay of a constant density due to the homogeneous boundary condition imposed at $x = 1$. The source injects particles in $x \leqslant 0.1$. The diffusion is non-linear; the diffusion coefficient $D \propto (\partial_x n)^2$. Hence, in regions where $n = \text{const.}$, $D = 0$. Therefore, in those regions, $\partial_t n = 0$; hence, no change in n. Initially, we expect the evolution to consist of a density increase for $x < 0.1$ (because of the source) and a decay near $x = 1$. For a time, the two regions (of decreasing and increasing density) are independent of each other since they are separated by a central region where $\partial_x n = 0$.

The numerical solution is displayed in Fig. 1 in which we plot $n(x)$ for several values of t. Curves 1, 2, etc., respectively, represent n for $t \approx 10^{-5}$, 10^{-4}, 10^{-3}, 10^{-2}, and 2×10^{-2}. For $t \leqslant 10^{-3}$, we see the non-linear diffusion of the cooling wave. Our choice of problem implies that the transient wave is self-similar, a phenomenon discussed in [12]. Indeed, if $\Gamma = -D_0(\partial_x n/n)^2\partial_x n$, by choosing the similarity variable,

$$\xi = (1/64D_0)x^4/t, \tag{16}$$

and examining the solution near $x = 1$, Eq. (5) is reduced to the ODE,

$$\xi v\left(12\xi\frac{\mathrm{d}v}{\mathrm{d}\xi} + 9v + 4\xi v^2\right) = -1, \quad v = \frac{1}{n}\frac{\mathrm{d}n}{\mathrm{d}\xi}.$$

The wave is characterized by the constant value of the wave front ξ_f. Inverting Eq. (16) gives the temporal dependence of the spatial position,

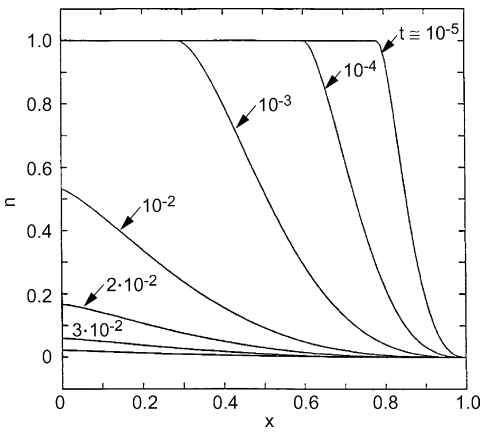

Fig. 1. Solutions $n(x,t)$ vs. x for analytic test problem, for several values of t.

$$x_f = \xi_f t^{1/4}.$$

Assuming the numerical solution to be correct at $t = 10^{-5}$, i.e., $x_f(t = 10^{-5}) \approx 0.215$ (when measuring from the right), we find $\xi_f = 3.339$. This implies the dependence of the front should be,

t	10^{-5}	10^{-4}	10^{-3}	10^{-2}
$1 - x_f$	0.785	0.618	0.321	−0.2

The position at $t = 10^{-4}$ and 10^{-3} agrees with Fig. 1. At $t = 10^{-2}$, the analysis no longer holds since by this time the cooling wave has interacted with the source region and the left-side boundary condition.

The effect of the source is not obvious from Fig. 1. At $t = 10^{-3}$, S has only raised the original density, $n = 1$, by the amount, $t \cdot S = 10^{-3} \cdot 1$ and this small amount is just barely visible in the figure. However, by $t = 10^{-2}$, the cooling wave has already interacted with the source. The source is most evident in the steady state solution,

$$n^e(x) = \frac{S_0 \delta}{27 D_0} \begin{cases} (L_x - x)^3 & \text{if } x < \delta, \\ \left[L_x - \delta + 0.75(\delta - x^{4/3}/\delta^{1/3}) \right]^3 & \text{otherwise.} \end{cases}$$

Without a source, all of n would flow out of the domain (let $S_0 = 0$ in the above equation).

2.2. Local approximation

The plasma turbulence model we use to illustrate our coupling technique is a numerical solution of the 2D (x, y) Hasegawa–Wakatani equations [6] for a perturbed density variable N_p and the z-component of the vorticity ζ:

$$\begin{aligned} &\partial_t \zeta + \nabla \cdot (\mathbf{v}\zeta) = \alpha(\phi - N_p) + \mu \nabla^2 \zeta, \\ &\partial_t N_p + \nabla \cdot (\mathbf{v} N_p) = \alpha(\phi - N_p) - \kappa \partial_y \phi + \nu \nabla^2 N_p. \end{aligned} \tag{17}$$

Here, the electric-field-drift velocity $\mathbf{v} = \hat{Z} \times \nabla\phi$, ϕ is the electrostatic potential normalized to T_e/e, the vorticity ζ is related to ϕ via $\zeta = \nabla^2\phi$, and $\kappa \equiv -\partial_x \log N_b$. Time is in units of inverse ion cyclotron frequency $\Omega_{ci} = eB/m_i c$, and distance is in units of $\rho_s \equiv c_s/\Omega_{ci}$, where the sound speed $c_s^2 = T_e/m_i$. The density variable N_p is related to the total physical density N and a specified stationary background N_b by:

$$N(x, y, t) = N_b(x)(1 + N_p(x, y, t)). \tag{18}$$

We note that N_p can itself have an average, and hence we separate it into its averaged and fluctuating parts:

$$N_p(x, y, t) = n(x, t) + \tilde{n}(x, y, t), \tag{19}$$

with $\langle \tilde{n} \rangle = 0$, where $\langle \cdot \rangle$ denotes the averaging operator. Of the various density variables just introduced, N must be positive, and we consider only positive N_b.

In 2D, the coupling coefficient $\alpha(x) = k_\parallel^2 T_e/m_e \nu_{ei} \propto 1/N_b$, where ν_{ei} is the electron–ion collision frequency and $k_\parallel$ is a chosen parallel wavenumber. As $\alpha \to \infty, N_p - \phi$ goes to zero; i.e., the electrons become fully adiabatic. In Eq. (17), the term $-\kappa\partial_y\phi$ is responsible for the growing modes, and is a source of energy (e.g., $\langle \phi^2 \rangle$) in the equations. For local calculations and ordinary profiles, $\kappa > 0$; in general, though, $\kappa(x)$ can be of either sign. The small diffusive terms proportional to μ and ν represent very fine-scale dissipative processes and are customarily included in simulations in order to remove energy from the highest wavenumbers,

where it arrives via mode-coupling. Without the μ and ν terms, energy accumulates at high wavenumbers; these terms are necessary to achieve (fluctuating) steady states.

Periodic boundary conditions in y are applied to the system, and averages are defined by

$$\langle f \rangle \equiv \frac{1}{L_y} \int_0^{L_y} \mathrm{d}y f. \tag{20}$$

Eq. (17) were originally developed by Hasegawa and Wakatani [6] and have been extensively studied, e.g., in [13]. We note here that they follow from the two-fluid Braginskii equations [14] by assuming a uniform, constant magnetic field, using drifts for perpendicular velocities (low-frequency approximation), neglecting electron inertia, assuming a spatially and temporally constant electron temperature, neglecting the ion parallel velocity, and assuming $N_p \ll 1$ but $\nabla_\perp N_b \sim \nabla_\perp N_p$.

In the local approximation, the fluctuations are assumed to respond only to the local macroscopic averaged fields and their x derivatives. A necessary condition for this to be a good approximation is that the radial (x) correlation length for the turbulence be short compared to the equilibrium scale length (as assumed in the derivation of Eqs. (17)). Hence a separation of scales has been formally carried out in x; a transport equation in the form of Eq. (5) evolves the background N_b with a flux Γ obtained by solving Eqs. (17). At each point that the flux is needed, an *independent* homogeneous turbulence calculation that knows *only* about the local value and gradient of N_b is performed. Thus, the x-domains of the transport and turbulence simulations differ, and to avoid confusion we introduce X and Y for the latter. In a given simulation of the turbulent Eqs. (17), α and κ are now spatial constants; boundary conditions are periodic in X as well as Y; solutions are translationally invariant, i.e., the turbulence is homogeneous; and averages are defined by

$$\langle f \rangle_L \equiv \frac{1}{L_X L_Y} \int_0^{L_Y} \mathrm{d}Y \int_0^{L_X} \mathrm{d}X f \quad \text{(local approximation)}. \tag{21}$$

Performing an X–Y average on Eqs. (17) shows that for the local version (uniform α, κ), due to periodicity in X, if $\langle n \rangle_L$ and $\langle \phi \rangle_L$ vanish at $t = 0$, they remain zero (though the instantaneous n and $\langle \phi \rangle$ are non-zero); then $\langle \xi \rangle_L$ must also vanish. Thus, the local version of Eqs. (17) provides a suitable candidate for coupling, as discussed in Section 1.

We now describe the details of coupling Eq. (5) to the code HAWC, described in [7], which we use to solve Eqs. (17) in this weakly varying-background limit. First note that HAWC makes use of a scale-invariance transformation that exists within the local approximation: if we transform to hatted variables $\hat{t} = \kappa t$, $\hat{x} = x$, $\hat{\phi} = \phi/\kappa$, $\hat{n} = N_p/\kappa$, and $\hat{\alpha} = \alpha/\kappa$, then κ drops out of the equations, and we see that there is only a one-parameter ($\hat{\alpha}$) family of solutions to Eqs. (17). (The coefficients μ and ν are kept small enough that they do not affect the solution away from the boundaries.) So we proceed with Eqs. (17) written in hatted variables, i.e., with $\kappa = 1$.

Eq. (5) is discretized into N zones. For each transport Δt, we run N *separate* HAWC simulations iteratively with the transport Eq. (5), and proceed until each of the HAWC simulations saturates. The HAWC runs are on a domain defined by the rectangular box

$$X_{\max} = Y_{\max} = 10\pi.$$

Each problem is given random but small initial conditions. Each HAWC problem has its own constant (in space) value of $\hat{\alpha}$; and at each iteration with transport (each HAWC time step $\Delta\hat{t}$) all the $\hat{\alpha}$'s are updated as follows. Defining $l_n \equiv -\kappa^{-1}$, for each transport mesh width (x_{j-1}, x_j), we compute from the transport's density profile n (i.e., N_b)

$$l_{n,j-1/2} = (1/2)(n_j + n_{j-1})(x_j - x_{j-1})/(n_j - n_{j-1}),$$

and then set $\hat{\alpha}_{j-1/2} = \alpha_{j-1/2} l_{j-1/2}$ as input to HAWC; here, $\alpha_{j-1/2} = (1/2)(\alpha_j + \alpha_{j-1})$, where $\alpha_j \propto 1/N_b(x_j)$.

The flux $\bar{\Gamma}$ actually used in the transport equation is obtained from the space average $\tilde{\Gamma} = -c_s N_b \times \langle N_p \partial_y \phi \rangle_L = -c_s N_b \kappa^2 \langle \hat{n} \partial_Y \hat{\phi} \rangle_L$ of the HAWC solutions. Defining $\tilde{D}_H = \langle \hat{n} \partial_Y \hat{\phi} \rangle_L$, we use

$$\bar{\Gamma}^l = -c_s N_b \frac{\bar{D}_H^{l-1}}{\bar{l}_n^{l-1}} \frac{\partial_x n^l}{n^{l-1}}, \tag{22}$$

where the bars on the right-hand side refer to the iteration-averaging scheme Eq. (11).

Before proceeding to a discussion of the results of the coupled-calculation simulations, we briefly explore the statistical and convergence properties of the uncoupled HAWC code.

In Fig. 2 we display a HAWC time history for $\hat{\alpha} = 0.1$, for 1000 time steps. Note that $\bar{D}_H(\hat{\alpha} = 0.1) \approx 1$ and that even for large l, the fluctuations vary between 0.5 and 1.5. This behavior is typical for other $\hat{\alpha}$. One important aspect of $\bar{D}_H(\hat{\alpha})$ (for the $\hat{\alpha}$ considered) is $\bar{D}_H > 0$. This is fortunate since $\bar{D}_H$ is the diffusion coefficient. However, for some $\hat{\alpha}$ we did see realizations for which $\tilde{D}_H^l(\hat{\alpha}) \leqslant 0$.

In the interest of speed, we chose to run HAWC for the coupled problems below, and in Fig. 2 above, on a limited 32×32 "grid" (HAWC is a pseudo-spectral code). If HAWC is run with enough resolution ($>64 \times 64$) throughout the range $0.01 \leqslant \hat{\alpha} \leqslant 1$, $\bar{D}_H$ can be approximated by the Padé fit

$$D_{\text{Padé}} = a_1 [\hat{\alpha}(1 + a_2 \hat{\alpha}^2)]^{-1/3}, \quad a_1 = 0.58, \ \ a_2 = 2.9. \tag{23}$$

This approximation has the expected behavior for large and small $\hat{\alpha}$, viz., if $\hat{\alpha} \ll 1$, $\bar{D}_H \propto \hat{\alpha}^{-1/3}$, and if $\hat{\alpha} \gtrsim 1$, $\bar{D}_H \propto \hat{\alpha}^{-1}$ [10].

The dependence of $\bar{D}_H$ on $\hat{\alpha}$ for the 32×32 runs is displayed in Fig. 3. Curve 1 is the average of 1000 realizations. Curves 2 and 3, are respectively, $\bar{D}_H \pm$ one standard deviation and curve 4 is Eq. (23). For $0.1 \leqslant \hat{\alpha} \leqslant 1.0$, curve 4 (Eq. (23)) is a poor fit to our low-resolution HAWC runs. A better analytic fit for this resolution is provided by another approximant,

$$\log_{10} D_A = -0.7 - 0.84\beta - 0.14\beta^2, \quad \beta = \log_{10} \hat{\alpha}. \tag{24}$$

We display Eq. (24) as curve 5 in Fig. 3. Except for the bump at $\hat{\alpha} \approx 0.04$, curve 5 is a very close fit to curve 1.

We now turn to the results of the coupled calculations.

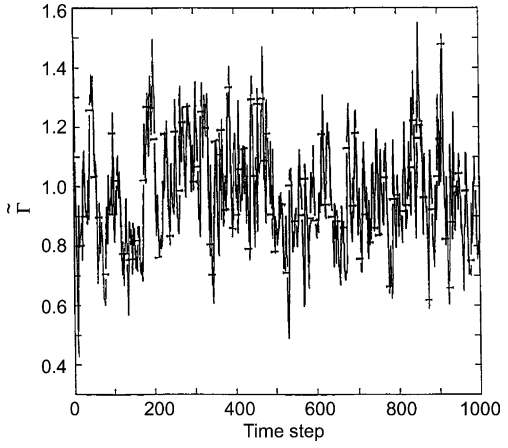

Fig. 2. Time step history of flux from HAWC.

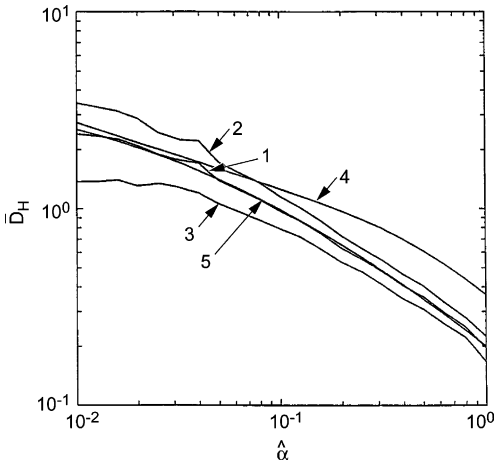

Fig. 3. Dimensionless diffusion coefficient $\bar{D}_H$ vs. $\hat{\alpha}$. Curve 1 is the average of 1000 realizations. Curves 2 and 3 are, respectively, $\bar{D}_H \pm$ one standard deviation. Curves 4 and 5 are fits given by Eqs. (23) and (24), respectively.

For the numerical test, except for setting $n^0 = 0.1$ and $c_1 = 0.001$ (see Eq. (7)), the initial and boundary conditions and the source are the same as in the previous section. The main difference between this problem and the one in Section 2.1 is that here the flux is given by Eq. (22).

The results for different t are displayed in Fig. 4. Note the similarity to Fig. 1. At $t \approx 0.016$, the cooling wave has reached the source region. The initial increase of n for small x is consistent with the source value and the elapsed time $S_0 t$. After $t \approx 0.016$, the source particles diffuse outwards until a steady-state is reached, curves 7 and 8.

To check the results, we make another run in which $\tilde{D}_H$ is replaced by D_A (see Eq. (24)). The result is nearly identical to the turbulence run. We present a comparison in Fig. 5 which plots the time history $n(x = 0, t)$ of both simulations. Curve 1 is the HAWC result and curve 2 corresponds to using D_A. For $t \leqslant 10^{-2}$, the curves exhibit an increase in n due to the source. The rise is followed by a decay to the

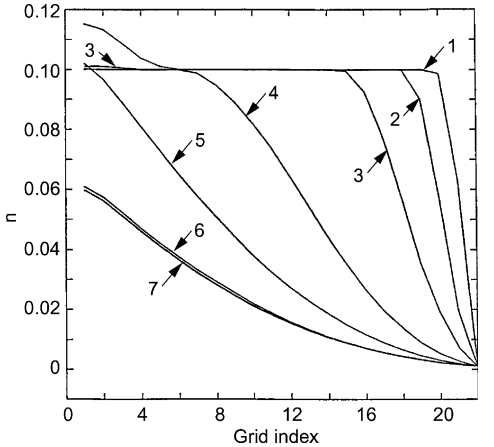

Fig. 4. Solutions $n(x, t)$ vs. x-grid index from HAWC for several values of t. Curves 1–7 correspond to, respectively, $t = 1.5 \times 10^{-5}$, 1.27×10^{-4}, 1.02×10^{-3}, 0.0164, 0.131, 1.05, and 16.8.

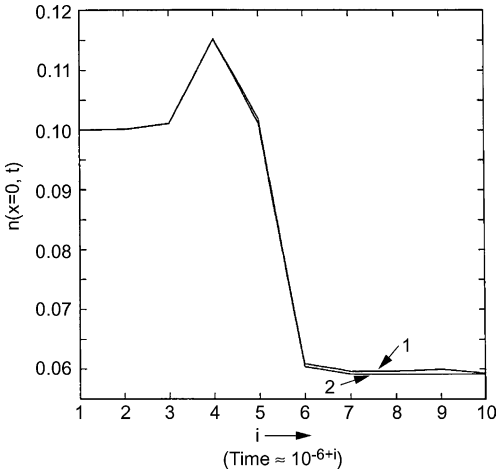

Fig. 5. Comparison of solution n at $x = 0$ vs. log(time) with flux obtained from HAWC (curve 1) with that obtained from fit Eq. (24) (curve 2). $i = 1, 2, \ldots, 7$ correspond to curves 1–7 in Fig. 4.

equilibrium value. This validates our technique for extracting the correct average from the fluctuating diffusion coefficient.

The turbulence code HAWC behaves best if $\hat{\alpha}$ is restricted to the range $0.01 < \hat{\alpha} \lesssim 1$. To comply, our calls to HAWC (to determine $\tilde{D}_H$) are made with the modified value,

$$\hat{\alpha}_H = \max(10^{-2}, \min(1., \hat{\alpha})).$$

The modification is of no consequence for the steady-state transport solution since the problem is designed so the equilibrium $\hat{\alpha}$ fall within the restricted range. However, the transient $\hat{\alpha}$ range over all positive numbers. To illustrate the differences and to demonstrate how our coupling scheme converges we present Figs. 6 and 7. Fig. 6 shows the l-iteration history at the transient time $t = 0.01638$ at $x = 0$. Curve 1 is $\hat{\alpha}$; note that it is >1. Curve 2 is $\hat{\alpha}_H$ and curve 3 is $5 \times \tilde{\Gamma}^l$. After 110 iterations, the averaged flux satisfies our

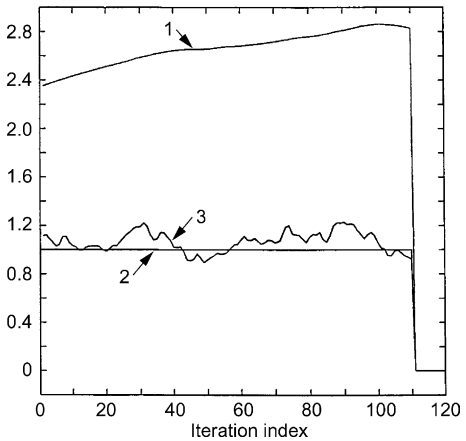

Fig. 6. Iteration history of $\hat{\alpha} = \alpha/\kappa$ (curve 1), limited parameter $\hat{\alpha}_H$ (curve 2), and five times the flux $\tilde{\Gamma}^l$ (curve 3). The iterations are within a transport time step at transient time $t = 0.01638$ (see Eq. (22)).

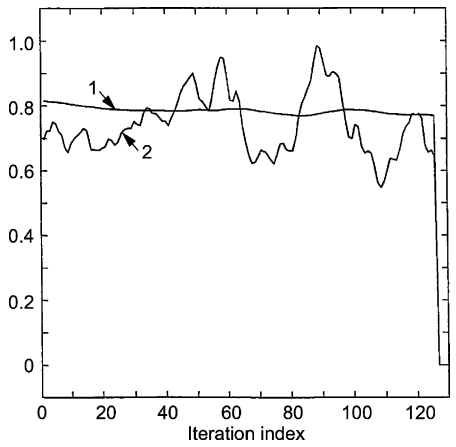

Fig. 7. Iteration history within a transport time step (see Eq. (22)) of $\hat{\alpha} = \alpha/\kappa$ (curve 1) and three times the flux $\tilde{\Gamma}^l$ (curve 2) at steady state ($t = 134.2$). At this time $\hat{\alpha}_H = \hat{\alpha}$.

convergence criterion and we move on to the next transport time step. The analogous steady-state iteration history is shown in Fig. 7 for which $t = 134.2$. Curve 1 is $\hat{\alpha}$; during the iteration its value hovers about 0.8; hence, $\hat{\alpha}_H = \hat{\alpha}$. Curve 2 is $3 \times \tilde{\Gamma}^l$; despite its fluctuations, the method homes in on the average.

3. Non-local coupling

In this section we describe the coupling between a transport equation and a non-local turbulent system, i.e., the general case outlined in Section 1. The turbulence is again modeled by Eqs. (17), but now the input parameters α and κ (the coefficients responsible for the growth of the turbulent eddies) are allowed to vary with x, in accordance with the density profile N_b from which they are defined. Formally, Eqs. (17) follow from a multiple-length-scale expansion of the Braginskii equations, with the parametric dependence on the (long) equilibrium spatial scale retained. In our numerical examples we will push these equations beyond their limits of validity by choosing situations where the equilibrium and fluctuation scale lengths are comparable and where $N_p \sim 1$ in order to test how well the coupling algorithms work under these circumstances. In these simulations our transport and turbulence equations are the averaged and fluctuating parts of the same primitive equations. In contrast with the previous section, non-local coupling requires only a single copy of the turbulence equations, solved on the same spatial domain as the transport equation. The turbulent medium is homogeneous in y, and spatial averages are as in Eq. (20):

$$\langle f \rangle \equiv \frac{1}{L_y} \int_0^{L_y} \mathrm{d}y\, f \quad \text{(non-local average)},$$

leaving quantities that are functions of x and t.

The main complication specific to coupling to non-local turbulence is that the turbulent eddies can assume a scale comparable to that of the equilibrium variations, implying a flux which is in general not a function of only local quantities. In particular, $\Gamma(x)$ need not necessarily flow down the gradient of the transported quantity. The thermodynamic principle one would derive from Eqs. (29) and (30) below would yield a single constraint, over the whole domain, on the direction of the net flow, and involve global integrals. If the flux is non-local (does not depend only on local fields and derivatives), we do not expect there to be any local thermodynamic principle. Stated another way, if the flux is non-local, it will respond to

properties of the profile over some range of positions and so need not flow everywhere down the gradient. This forces us to modify our scheme to allow for anti-diffusive behavior. The modifications are in keeping with certain desired properties of the numerical solution: that it remain positive and that it be conservative. We accomplish this by expressing Γ as a linear combination of ∇n and cn where c plays the role of a convective velocity.

Consider a generic transport equation for a "density" f,

$$\frac{1}{g}\partial_t f + \partial_x \Gamma = S - of, \tag{25}$$

where $g(x)$, $o(x)$, and $S(x,t)$ are known functions. In order to avoid any Δt stability restrictions, fully implicit time stepping is used. If m denotes the time level, we write

$$(f^m - f^{m-1})/g\Delta t + \partial_x \Gamma^m = S^m - of^m. \tag{26}$$

As before, we use linear finite elements and lump all but the transport term, e.g.,

$$\int \mathrm{d}x\, \phi_j f^m/g \to f_j^m \left[(h/g)_{j-1/2} + (h/g)_{j+1/2}\right]/2,$$

where $h_{j-1/2} = x_j - x_{j-1}$ is the mesh width. As for the local case, in order to obtain a self-consistent advanced-time step flux, Eq. (26) is solved by iterating it with the temporal steps of the turbulence code. For the non-local coupling, we use the code HAWCX, described by Xu et al. [8]. If l is an iteration index, $f^{m,l}$ is the unknown quantity. Henceforth, we omit the time index m. In addition to f^l, we also keep track of a running average $\bar{f}^l$, defined as in Eq. (11) (or alternatively, (12)). The average has dual use. First, the gradient of $\bar{f}^l$ is used to compute the modifications to HAWCX's coefficients. Secondly, since f^l solves a linear system, self-consistency (convergence of the iterations) requires that $\bar{f}^l$ satisfy the linear system to some tolerance.

The iteration cycle begins by defining the averaged density gradient that drives the turbulence (see Eq. (33) below), using $\bar{f}^{l-1}$ or f^{l-1}. HAWCX returns with the next noisy iterate for the flux $\tilde{\Gamma}^l$. This iterate is folded into an average flux $\bar{\Gamma}^l$ according to Eq. (12). In the results, we iterate at least 2000 times and use a window size $N \geqslant 500$. Proper choices for N and the total number of iterations allowed are problem dependent. If the turbulence code uses a time step which is small compared to the characteristic time scales (linear growth times, characteristic wave periods, and decorrelation times), then successive $\tilde{\Gamma}$ iterates differ little from each other and very many l-iterations are required in order for the turbulence code to saturate (this is the case for HAWCX). It is also difficult to determine when HAWCX has saturated since there are often waves with low frequencies traveling across the domain – even after saturation.

Once $\bar{\Gamma}^l$ is known, it is used to define zone-centered "diffusion" and "velocity" coefficients,

$$D_{j-1/2}^{l-1} = -\bar{\Gamma}_{j-1/2}^l / d_x f_{j-1/2}^{l-1} \quad \text{and} \quad c_{j-1/2}^{l-1} = \bar{\Gamma}_{j-1/2}^l / f_{j-1/2}^{l-1}. \tag{27}$$

D and c are the coefficients that would result if the turbulence-code flux were represented as fully diffusive or fully convective, respectively; for use in the transport-code, we will use a weighted combination, to be described shortly below, of these two representations, with the weight chosen for numerical convenience and the combination guaranteed (at the end of the time step) to sum to the turbulence-code's flux. We emphasize that there need be no connection between the weight or either of these representations and an analytic expression for the dependence of the turbulent flux on the fields; if the flux in fact depends, for example, on higher derivatives or on integrals of n, this does not result in an error in our transport-equation solution.

The zone-centered $d_x f$ and f terms in Eq. (27) are obtained from the nodal values $\bar{f}_j^{l-1}$

$$d_x f_{j-1/2}^{l-1} = (\bar{f}_j^{l-1} - \bar{f}_{j-1}^{l-1})/h_{j-1/2} \quad \text{and} \quad f_{j-1/2}^{l-1} = (\bar{f}_j^{l-1} + \bar{f}_{j-1}^{l-1})/2.$$

We could also define an upwinded velocity by looking at $\text{sign}(\bar{\Gamma}_{j-1/2}^{l})$ and let $f_{j-1/2}^{l-1} = \bar{f}_j^{l-1}$ if the flux is negative. The D and c coefficients *actually used* should be limited to keep the problem well posed for the numerical linear transport-equation solver. For example, assume that $D_{\min}$ and $D_{\max}$ are two non-negative numbers limiting the diffusion, and let

$$\theta_{j-1/2} = \begin{cases} 0 & \text{if } D_{j-1/2} < D_{\min} \text{ or } D_{j-1/2} > D_{\max}, \\ (D_{\max} - D_{j-1/2})/(D_{\max} - D_{\min}) & \text{otherwise.} \end{cases}$$

Clearly, $0 \leqslant \theta \leqslant 1$. It is easy to let θ have a non-linear dependence on D or to prescribe a range for D for which $\theta = 1$. We then use θ to apportion the flux into diffusive and convective terms according to

$$\Gamma_{j-1/2}^{l} = -\theta_{j-1/2} D_{j-1/2}^{l-1} d_x f_{j-1/2}^{l} + (1 - \theta_{j-1/2}) c_{j-1/2}^{l-1} f_{j-1/2}^{l}. \tag{28}$$

There are many other ways to represent the flux used in a transport code, and within the choice (28) there is wide latitude in the selection of $D_{\min}$ and $D_{\max}$. Clearly, the D actually used in the transport code (in our case, $\theta_{j-1/2} D_{j-1/2}^{l-1}$) must be non-negative and bounded, whereas $D_{j-1/2}^{l-1}$ will go through infinity at an extremum of n (thus the need for a finite $D_{\max}$) and be locally negative should the flux be running locally up hill; we also found empirically that limiting $D_{\max}$ can improve the stability of the iterations within a time step. In Section 3.1 and in Appendix A we present two other methods for ensuring a well-posed problem for the linear diffusion-equation solver.

In Eq. (28) we have again used the concept of a zone-centered f. By introducing another zone-centered coefficient, we write

$$f_{j-1/2}^{l} = \beta_{j-1/2} f_{j-1}^{l} + (1 - \beta_{j-1/2}) f_j^{l}, \quad 0 \leqslant \beta \leqslant 1.$$

The β coefficients allow us to guarantee necessary properties of the nodal values f_j^l. Their choice is driven by the desire to have an M-matrix which in this case requires that we load only non-negative elements on the matrix diagonal and only non-positive contributions to the off-diagonals.

Ignoring the boundary term, a straightforward integration by parts of the flux term gives (see definition of ϕ above Eq. (14)),

$$\int \mathrm{d}x\, \phi_j(x) \partial_x \Gamma = -\int \mathrm{d}x\, \phi_j' \Gamma = -\Gamma_{j-1/2} + \Gamma_{j+1/2}.$$

According to Eq. (28), the diffusive part of $-\Gamma_{j-1/2}$ is

$$(\theta D^{l-1}/h)_{j-1/2} (f_j^l - f_{j-1}^l),$$

while the convective part of $-\Gamma_{j-1/2}$ is

$$-(1 - \theta_{j-1/2}) c_{j-1/2}^{l-1} \left[\beta_{j-1/2} f_{j-1}^{l} + (1 - \beta_{j-1/2}) f_j^{l} \right].$$

This leads to the proper choice for β. If the flux is purely diffusive ($\theta = 1$), then β is of no consequence. If the flux is purely convective, then

$$\text{if } \theta = 0, \quad \beta = \begin{cases} 1 & \text{if } c \geqslant 0, \\ 0 & \text{if } c < 0. \end{cases}$$

This choice is equivalent to implicit upwind differencing. If $0 < \theta < 1$, then β is allowed to creep towards 1/2 from below (above) if $c < (\geqslant) 0$. The creep is not allowed to go to such a level that when the diffusive and

convective terms are combined, they create a positive contribution to the off-diagonals or a negative contribution to the diagonal. This assures an M-matrix; it is tridiagonal but no longer symmetric. If $M = LU$ is the decomposition and $y \geqslant 0$, by examining the elements of L and U, it is easy show that the solution of

$$Mf^l = y$$

satisfies $f^l \geqslant 0$.

3.1. Formulation of non-local coupling test problem

Here we derive the coupled equations used for the non-local coupling studies. We note that Eqs. (17) lead to an evolution of the averaged quantities $\langle N_p \rangle$ and $\langle \zeta \rangle$. (This is in spite of the implication in the derivation [6] that N_p and ζ are perturbations from averaged quantities.) We shall work with a modified version of Eqs. (17) in which there is no evolution of $\langle \zeta \rangle$, but $\langle N_p \rangle$ still evolves. This modified system provides a numerical test of our coupling algorithm, as it can be solved in two ways: (1) direct solution using the turbulence code HAWCX; and (2) a coupled solution, in which Eq. (17) is further modified so that neither $\langle N_p \rangle$ nor $\langle \zeta \rangle$ evolves, and the equation for the evolution of $\langle N_p \rangle$ is solved by the transport code. Analogously to Eq. (19), we introduce the averaged and fluctuating parts of the potential and velocity:

$$\phi(x,y,t) = \langle \phi \rangle + \tilde{\phi},$$

$$\zeta(x,y,t) = \langle \zeta \rangle + \tilde{\zeta},$$

where the averages $\langle \cdot \rangle$, defined in Eq. (20), are functions of x and t.

Similarly, Eqs. (17) are themselves separated into two systems, one for the averaged variables n, $\langle \phi \rangle$, and $\langle \zeta \rangle$, and another for the fluctuating variables $\tilde{n}$, $\tilde{\phi}$, and $\tilde{\zeta}$. The two systems are coupled in a manner similar to that of the local case. "Fluxes" of the averaged variables are determined by $\tilde{n}$ and $\tilde{\phi}$ while certain coefficients of the fluctuating system depend on gradients of n.

In Eq. (17), we replace the density diffusion term by $\partial_x[v\partial_x(N_p - n)]$, where $v(x)$ is defined below, and we set $\mu = 0$. (The HAWCX code has intrinsic dissipation in its vorticity equation due to its differencing scheme.) Our modified version of Eqs. (17), which does not evolve $\langle \zeta \rangle$, is then given by subtracting the averaged ζ equation from the ζ equation itself and substituting $\tilde{\zeta} \to \zeta$, to yield

$$\partial_t \zeta + \nabla \cdot (\mathbf{v}\zeta) - \partial_x \langle v_x \zeta \rangle = \alpha(\phi - \tilde{n}), \tag{29}$$

$$\partial_t N_p + \nabla \cdot (\mathbf{v} N_p) = \alpha(\phi - N_p) - \kappa \partial_y \phi + \partial_x[v\partial_x(N_p - n)]. \tag{30}$$

In writing Eq. (29), we assumed the initial condition $\langle \phi \rangle = 0$; Eq. (29) implies that $\langle \zeta \rangle$ and $\langle \phi \rangle$ are both zero for all time, and then $\mathbf{v} = \tilde{\mathbf{v}}$.

If the equation for N_p is averaged, one obtains (recall definitions of N_p, n and $\tilde{n}$, Eqs. (18) and (19)):

$$\partial_t n + \partial_x \Gamma = -\alpha n \quad \text{with } \Gamma = -\langle \tilde{n} \partial_y \phi \rangle. \tag{31}$$

Subtracting this from the equation for N_p yields the evolutionary equation for $\tilde{n}$:

$$\partial_t \tilde{n} + \nabla \cdot (\mathbf{v}\tilde{n}) + \partial_x \langle \tilde{n} \partial_y \phi \rangle = \alpha(\phi - \tilde{n}) - (\kappa - \partial_x n)\partial_y \phi + \partial_x(v \partial_x \tilde{n}). \tag{32}$$

In Eq. (31), the flux is an average of the product of fluctuating variables. In Eq. (32), $\partial_x n$ modifies the coefficient responsible for the growth of the turbulence, i.e.,

$$\kappa \to \kappa' \equiv \kappa - \partial_x n. \tag{33}$$

Eq. (31) is the desired transport equation; its flux cannot be assumed to be of opposite sign to ∇n, either initially or at saturation. For small times, $n \approx 0$, and Γ tends to move down the gradient of N_b, which, over a large portion of the domain, need not be in the same direction as the gradient of n. As a further complication, inasmuch as n is not a physical density it can take on both positive and negative values. Nevertheless, we show below that our technique correctly computes its saturated value.

The validation consists of comparing the saturated value of n obtained by two methods, one the standalone or S way, another the coupled or C way. For the former, we run HAWCX by itself (solving Eqs. (29) and (30)), wait until the results saturate, and compute n. In the C way, HAWCX solves Eqs. (29) and (32), i.e., without evolving n. Instead, n evolves via Eq. (31) which is advanced by the scheme described in the previous section. Each iteration of the transport requires a time advancement of Eqs. (29) and (32) where κ' is modified by the transport according to Eq. (33).

Since the "density" n violates the positivity requirement of our transport solver, instead of Eq. (31) we solve an equivalent equation in which n is replaced by another variable which remains positive. For example, if we expect n to be bounded from below for all t,

$$n(x,t) > -d, \quad d > 0,$$

then we might choose to define

$$n^+(x,t) = n(x,t) + d$$

and obtain the modified transport equation,

$$\partial_t n^+ + \partial_x \Gamma = \alpha(d - n^+). \tag{34}$$

Since $\partial_x n^+ = \partial_x n$, κ' and Γ are unchanged. In terms of n^+, the physical density $N = N_b \times (1 + \tilde{n} + n^+ - d)$.

This choice, however, does not address the problem of flux running up the gradient of n. Hence, we define an alternate change of variables in order to be able to express Γ as a Fick's law, i.e., as $\Gamma = -D\nabla N^+$ with $D \geqslant 0$. Recall that in the original derivation of Eqs. (17) [6], the total density N is expressed as a product of a fixed background and a perturbation about that background, as in Eq. (18):

$$N(x,y,t) = N_b(x)(1 + N_p(x,y,t)).$$

The derivation assumes $|N_p| \ll 1$. However, in order to obtain a stringent numerical test, we choose an $N_b(x)$ that leads to an N_p that violates this condition. In order to recover both a positive density and a positive D, we define a new density N^+ analogously to n^+ above. Then, using Eq. (18), n is given by

$$n(x,t) = \frac{N^+(x,t) - d}{N_b(x)} - 1, \quad \text{where } N^+ = \langle N(x,y,t) \rangle + d. \tag{35}$$

This leads to the second modified transport equation,

$$\frac{1}{N_b(x)} \partial_t N^+ + \partial_x \Gamma = \alpha + \frac{\alpha}{N_b(x)}(d - N^+). \tag{36}$$

For Eq. (36), at least in the early stages, we expect Fick's law to hold. That is, for the same Γ as before, $-\Gamma/\partial_x N^+ \geqslant 0$. This modification is an example of adding a simple, constant in space and time, alias function to the density. Later in time, if the flux runs uphill, we treat Γ by the convective/diffusive split of Eq. (28). An alternate way of ensuring that D is positive, which can be applied to either Eq. (34) or (36), is to add a d which is a function of x, as described in Appendix A .

Finally, we discuss the boundary conditions and the dissipative function v. In HAWCX, the "hard-wired" conditions are

$$\zeta, \phi, \tilde{\boldsymbol{n}} = 0 \quad \text{at } x = 0, L_x$$

and periodicity in y. In addition, HAWCX's difference scheme requires ghost-point values. If (-1) is the ghost-point index to the left of $x = 0$, HAWCX sets

$$f_{-1} = -f_1,$$

where f denotes ζ, ϕ, or $\tilde{\boldsymbol{n}}$. For the transport, the boundary conditions depend on which of Eqs. (34) or (36) we use. In either case, we fix the density at the ends. For Eq. (34) we set $n^+ = d$ at $x = 0, L_x$, while for Eq. (36) we set,

$$N^+(0,t) = N_b(0) + d \quad \text{and} \quad N^+(L_x,t) = N_b(L_x) + d.$$

The dissipative function v is introduced to avoid complications at the boundaries; see Section 3.2. Its effect is to add diffusion at the endpoints but not disturb the evolution in the interior. In the examples,

$$v(x) = v_0[\exp(-x^2/\Delta_v^2) + \exp(-(x - L_x)^2/\Delta_v^2)]. \tag{37}$$

3.2. Numerical results

We compare results for the two ways of reaching a saturated state, the stand-alone or S way and the coupled or C way – see Section 3.1. For the tests, we choose a background profile $N_b(x)$ and then let the turbulence evolve to a saturated state. Two different profiles are used. The first is symmetric about the middle of the domain, which creates an anti-symmetric flow. For the second profile, $N_b' < 0$ throughout the domain and consequently the average flow is to the right. In the S runs, HAWCX is run for many cycles (time steps); frequently more than 10,000 are required to saturate. Saturation is checked by saving successive averages. Typically, in the S runs, HAWCX is run for batches of 1000 cycles each. After each cycle, the result is accumulated in a separate variable, and the batch average is obtained by dividing by 1000. Sometimes the batch averages differ, even after saturation. This implies that waves with long periods are still present. The "saturated" variable displayed in the figures is either the total averaged density $\langle N \rangle$ or the average of the perturbed part n.

When the S run saturates, it does so with respect to the fixed background N_b. For the C runs, the fully implicit differencing scheme does not impose any stability restriction on Δt. In fact, setting Δt to an arbitrarily large value reduces Eq. (26) to the desired discretization for the steady state. Since HAWCX's time step $\Delta\tilde{t}$ is usually $<$O(1), we set the transport time step Δt to 10^{10}. This not only allows us to compare the saturated states, but in some cases facilitates reaching the state "faster", where the efficiency is measured in the number of cycles that HAWCX requires to saturate. In the C runs, the transport computes n while HAWCX evolves $\tilde{n}$. We believe the time scale separation to depend on the ratio of the original gradient scale of N_b to the normalizing distance ρ_s (defined below Eq. (17)). In the C runs, we usually iterate for 2000 cycles, i.e., call HAWCX 2000 times, each time asking it to advance one $\Delta\tilde{t}$. If 2000 cycles are not enough for saturation, we call it again for another 2000 cycles, etc. This allows us to quote a time or equivalently an iteration number for the comparisons. For example, we can compare an S-run average of 1000 cycles after cycle 10,000 with a C-run average after cycle 4000.

In the runs we use $N_x = 60$ as the number of transport grid points. Hence, the transport, which computes node-centered values and has Dirichlet boundary conditions, has N_x unknowns. Since HAWCX returns a flux, which for the transport is zone centered, it solves its equations on a staggered grid. Since $N_x + 1$ fluxes

are required, HAWCX has that many unknown grid points in the x direction while it uses $N_x + 4$ points in the (periodic) y direction. The plots use the grid index for the abscissa.

The primary effect of the saturated state is to flatten the total density $\langle N \rangle$. This is evident by comparing $N^+ - d$ with N_b. The v function acts as a diffusion coefficient for $\tilde{n}$ and allows the saturated profiles to be less dependent on HAWCX's hard-wired boundary conditions. Simulations with $v = 0$ do not saturate easily. The locations of the large gradients of N^+, which initially coincide with those of N_b, later shift toward the boundaries. Setting v large near the endpoints relieves the difficulties. The function is initialized as follows: recalling Eq. (37), we set $v_0 = 5$ and $\Delta_v = 5L_x/N_x$. Hence, 5 (10) grid points away from the boundary, v decreases by a factor of e (e^4).

For the first comparison, we choose N_b symmetric about the middle of the domain; see Fig. 8, curve 1. The smooth transition between the ends where $N_b = 0.5$ and the middle where $N_b = 1.0$ is a cosine. Note that N_b has regions where it is completely flat. For this problem we set

$$\alpha(x) = 10^{-4}/N_b(x) \quad \text{and} \quad \kappa(x) = N_b'(x)/N_b(x).$$

The physical domain is,

$$L_x = L_y = 20\pi.$$

The average flow is anti-symmetric about $L_x/2$. Curves 2 and 3 of Fig. 8 are the C results for N^+ after cycle 200 and 2200, respectively. Curve 2 is the instantaneous result and curve 3 is the average over the previous 2000 cycles. (The average after only 200 cycles – which is not shown – lies much higher than the instantaneous result since the former contains remnants of the initial, low-amplitude transients.) We apply Eq. (12) with $N = 500$ to compute the average. The close agreement between curves 2 and 3 shows that the transport saturates after only 200 calls to HAWCX. For this example, we use Eq. (36) to model the transport. Since there is no threat of a negative N, we set $d = 0$.

The y-averages of the perturbed part, n and $\bar{n}$, i.e., the instantaneous and time-averaged density perturbations, are compared in Fig. 9. Curves 1 and 2 display the C results. Those curves respectively correspond to curves 2 and 3 of Fig. 8. Curve 3 of Fig. 9 is the S result after cycle 20,000 (average using Eq. (11) over cycles 19,000–20,000). Except for the slight discrepancy near $j = 10$, the runs agree. However, the S run takes a remarkably long time to saturate. In Fig. 10, curves 1, 2, 3 and 4, we present the S averages after

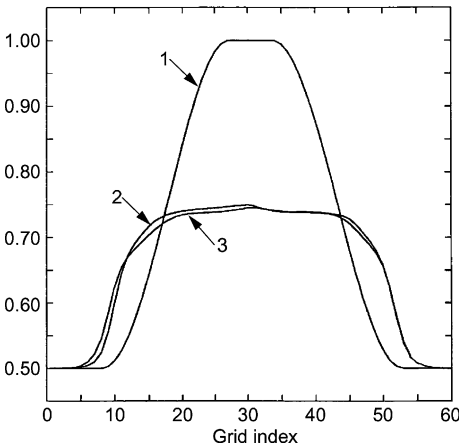

Fig. 8. Densities vs. x-grid index for coupled HAWCX-transport simulations. Curve 1 is the input background density N_b. Curve 2 is the instantaneous result for N^+ after 200 cycles; curve 3 is the average (over 2000 cycles) of N^+ after 2200 cycles. Alias density $d = 0$.

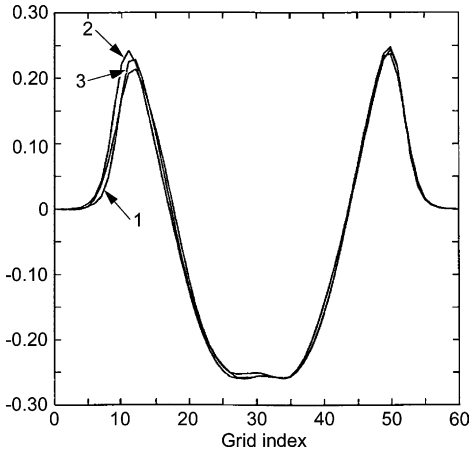

Fig. 9. Instantaneous and time-averaged coupled-simulation results (curves 1 and 2, respectively) for the density perturbation compared to the result of a stand-alone HAWCX simulation (curve 3), plotted vs. x-grid index.

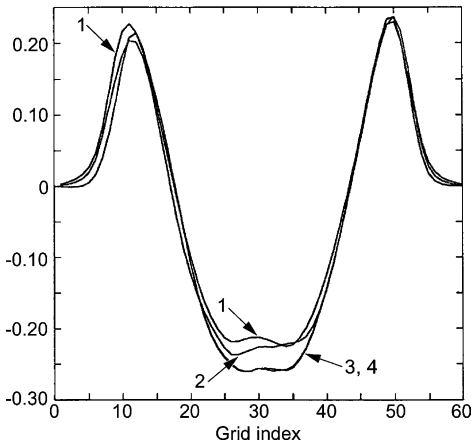

Fig. 10. Stand-alone HAWCX simulation results for same problem as Fig. 9. Curves 1, 2, 3, and 4 are averages taken after cycles 1000, 10,000, 15,000, and 20,000, respectively, plotted vs. x-grid index.

cycle 1000, 10,000, 15,000, and 20,000. Note how long it takes the S runs to saturate fully in the middle of the domain.

The comparisons are surprisingly good in light of the difficulty the transport encounters when computing the effective diffusion coefficient, Eq. (27), in the flat regions where $d_x f = 0$.

In the second example we choose N_b as follows:

$$N_b(x) = 1. - N_1 \tanh \frac{x - L_x/2}{\Delta_N}, \tag{38}$$

where $N_1 = 0.8$, $\Delta_N = 20.0$, and $L_x = 80\pi$.

For this problem we set

$$\alpha(x) = 2.5 \times 10^{-5}/N_b(x) \quad \text{and} \quad \kappa(x) = N_b'(x)/N_b(x).$$

We present C results for two different versions of the transport equation, Eqs. (36) and (34). Results for Eq. (36) are subscripted by 1, for Eq. (34) subscripted by 2. When using Eq. (36), we set $d = 1$, while for Eq. (34), we use $d = 2$. Thus, the subscript denotes the constant required to keep the variable positive. In the figures, d is subtracted from the results.

In Fig. 11, curve 1, we display $N_b(x)$; curves 2, 3, and 4 are averages of $N_1^+ - d$ after 8200, 10,200, and 12,200 cycles. The discrepancy between the curves reflects the statistical noise-level of the saturated state. The saturated state is peculiar. The steep gradient of N_1^+ undergoes a sign reversal in the middle of the domain. Inspection of the numerical results shows that the average flux is positive everywhere. This problem takes a long time to saturate. The early perturbations occur in the central region where N_b has the sharp gradient. As in the previous problem, the initial tendency is to flatten N_b by moving mass from the high- to the low-density side. However, the flattening of the central region broadens the total profile and extends the perturbation out to the boundaries. At the boundaries (especially the left one) the growing perturbation conflicts with the imposed boundary condition, $n = 0$. The introduction of dissipation, Eq. (37), alleviates the problem by keeping the perturbation away from the boundary. We speculate that this state is not physical but instead indicates usage of Eqs. (17) in an inappropriate regime. Indeed, the total density is occasionally negative. In Fig. 12 we compare the y-averages of the perturbations. Curves 1 and 2 correspond to n_1^+; they are derived from curves 2 and 3 of Fig. 11. Curves 3 and 4 of Fig. 12 correspond to n_2^+. There is some disagreement, but all curves share the salient features of the saturated state: it moves a great deal of density from the left half of the domain to the right. Fig. 13 compares the final results. Curve 1 displays $n_1^+ - d$ after 20,000 cycles while curve 2 displays $n_2^+ - d$ after 20,000. Curve 3 is the S result after 30,000 cycles averaged over the prior 1000 cycles. The S-simulation was nearing but had not yet achieved saturation at 30,000 cycles (an S-run differing in computational as opposed to physical parameters, which shows the approach to saturation, is presented in Appendix A).

The good comparison in Fig. 13 both demonstrates the numerical advantage and validates the robustness of our technique. This is a hard problem. Non-local effects are strong, as indicated by the fact that the turbulent eddy size is comparable to the background gradient scale-length. In part of the domain, the flux flows up the gradient of both the total density N^+ and the perturbation n. Nevertheless, our method is robust enough to find the saturated value. Its implicit nature is not restricted by the size of Δt.

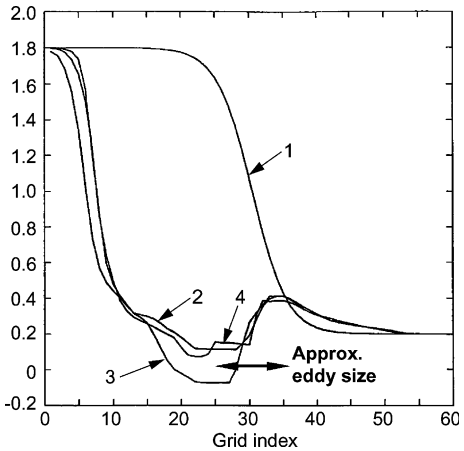

Fig. 11. Coupled-simulation results for input background density N_b (curve 1). Curves 2, 3, and 4 are averages of $N_1^+ - d = \langle N \rangle$ (see Eq. (35)) after 8200, 10,200, and 12,200 cycles. The curves are plotted vs. x-grid index.

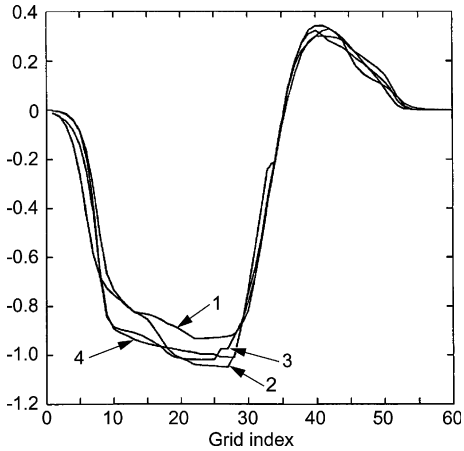

Fig. 12. Comparison of coupled-simulation results for two formulations of the transport equations. Curves 1 and 2 plot $n_1^+ - d$, derived from curves 2 and 3 of Fig. 11. Curves 3 and 4 are the corresponding results for $n_2^+ - d$, the solution of Eq. (34). The curves are plotted vs. x-grid index.

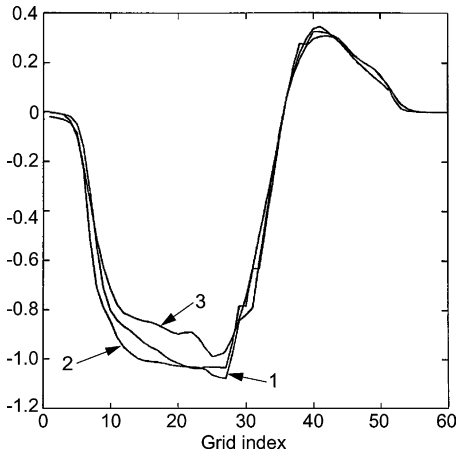

Fig. 13. Solutions at the final cycles of the simulations of Figs. 11 and 12. Curves 1 and 2 display $n_1^+ - d$ and $n_2^+ - d$ after 20,000 cycles. Curve 3 is the stand-alone result, just approaching saturation at 30,000 cycles. The curves are plotted vs. x-grid index.

4. Conclusions

We have described a very promising method of coupling transport and turbulence, which we illustrate with an application to the 2D Hasegawa–Wakatani model equations for plasma turbulence. We have augmented the basic scheme with two methods, about equally successful, for dealing with locally anti-diffusive behavior. Both local and global implementations of the coupling work well. For the problems where there is at least a moderate separation of spatial scales between the fluctuations and the background, the coupled approach achieves significant savings over the comparison stand-alone simulations, whether or not global effects emerge, while finding, to within expected statistical variations, the same averaged profiles.

We note that for the parameters used in our simulations and for parameters of typical interest for our motivating application (drift-wave-type turbulence in tokamaks), the global implementation of the

coupling (where a single large turbulence code with the same radial (x) domain as the transport code is used) is more efficient than the local one (where separate copies of the turbulence code are used at each transport mesh-point). This will be true (assuming a linear scaling in computational time with the number of radial mesh points for the turbulence codes) so long as $L^{\mathrm{tr}}/L^{\mathrm{fluc}} < N_x^{\mathrm{tr}} \times N^{\mathrm{fluc\_local}}$. Here L^{tr} and L^{fluc} are the background and fluctuation spatial scales respectively; N_x^{tr} is the number of radial mesh points for the transport code; and $N^{\mathrm{fluc\_local}}$ the number of eddies in the radial direction per local turbulence simulation necessary for convergence.

To do couplings for physical problems of the greatest interest (including the application to tokamaks, which inspired this work), we have to address the coupling of multiple fields. Here off-diagonal contributions of the gradients to the fluxes, which can be significant and of either sign, and whose analytic form in general will not be known, can have a serious impact on the stability of the method. Stability of the coupling algorithm near critical gradients (that is, values of the gradients at which the turbulence-driven diffusion coefficient vanishes) is also a potential issue.

Encouraged by the results obtained and presented in this paper (the work was carried out in 1995), we proceeded to couple a "full-physics" tokamak-core transport code to what was then the state-of-the-art 3D tokamak-core local fluid turbulence code, GRYFFIN [15,2]. Details of the two codes' preparation and communication, modifications to the algorithms presented in this paper, and results are given in [16]. The conclusions of the work, though preliminary, included: (1) few refinements of the schemes presented in the present paper were needed; (2) an extension of the coupling to two fields, electron and ion temperatures, succeeded (although normally tightly coupled by drag, we simulated cases with a significant difference driven by the large ion heat-flux returned by GRYFFIN); (3) global coupling is probably more robust than local, because neighboring regions in a global code influence each other directly and so tend to return self-limited profile gradients.

Appendix A. Adaptive alias

In this appendix we describe an alternate method of handling the problem of fluxes running locally up hill (or other situations where the diffusion coefficient locally is outside acceptable bounds). The basic approach is to add an "alias density" n_{al} to the density variable of interest, n, in such a way that the flux Γ runs down the hill of the total density $n_{\mathrm{tot}} = n + n_{\mathrm{al}}$ with a diffusion coefficient $D = -\Gamma/\partial_x n_{\mathrm{tot}}$ which is within the acceptable bounds.

The inspiration for this approach comes from the definition of the various densities in the non-local Hasegawa–Wakatani turbulence model of Section 3.1. We recall that the physical density N is related to the Hasegawa–Wakatani density variable N_p by the relation $N = N_b(1 + N_p)$. From the examples shown in Section 3.2, it is apparent that even for relatively benign situations where the flux flows down the gradient of N, it can flow up local gradients of N_p. If we were to construct a "transport code" for N_p we would encounter the "flux running up hill" problem, whereas the multiplication by N_b and the addition of the fixed density N_b results in a density whose gradient more closely defines the direction of Γ.

The alias procedure would be simple if we had some reason to a priori know the direction of the flux; we could just add to the physical density a fixed alias function whose gradient is everywhere opposite to the expected flux, and whose magnitude is so large that the sum of the physical and alias gradients also has the property that the flux runs down the hill of the total gradient. While, for transport in a tokamak, the transport of particles is typically down a density gradient from the center to the edge, this need not always be the case (for example, if there is a hollow density profile and the flux remains outward). To account for this possibility, we introduce an alias function which can change at every iteration. The alias is defined by the following procedure.

Starting from the latest estimate of the turbulent flux Γ and of the total density $n_{\rm tot}$ (which is just the actual density n being evolved by the transport equation if an alias density has not been previously defined) we define an initial array of diffusion coefficients by $D_j = -\Gamma_j/(\mathrm{d}n_{\rm tot}/\mathrm{d}x)_j$. Some of the members of this array might be outside of the acceptable range of diffusion coefficients. Hence, we define a mapping that takes each value of D to a new value $D_{\rm new}$ which lies within the acceptable range of diffusion coefficients. For example, if we demand that $D_{\rm min} < D_{\rm new} < D_{\rm max}$, then a reasonable choice for the mapping is given by

$$D_{\rm new} = \begin{cases} D_{\rm min}, & |D| \leqslant D_{\rm min}, \\ |D|, & D_{\rm min} < |D| < D_{\rm max}, \\ D_{\rm max}, & |D| \geqslant D_{\rm max}. \end{cases} \tag{A.1}$$

We then define a new density-gradient array $(\mathrm{d}n_{\rm new}/\mathrm{d}x)_j$ from the new diffusion coefficient,

$$(\mathrm{d}n_{\rm new}/\mathrm{d}x)_j = -\Gamma_j/D_{{\rm new}_j} \tag{A.2}$$

and hence a new density variable which follows from the relation

$$n_{{\rm new}_j} = n_{{\rm new}_{j-1}} - \Delta x \Gamma_j/D_{{\rm new}_j} \tag{A.3}$$

This relation may be used to integrate from either end of the spatial domain; we alternate directions in succeeding iterations in order to help ensure the stability of the aliasing procedure (avoiding a run-away alias). In either case, we start the integration by assuming that $n_{\rm new}$ equals the old value at the starting end. In a similar vein, at the end of each sweep, we add a constant to the array of $n_{\rm new}$ values to make the minimum $n_{\rm new}$ equal a specified small value. Once $n_{\rm new}$ is known, the new alias function follows from $n_{\rm al} = n_{\rm new} - n$ (and n is unchanged during the re-definition of the alias function).

We performed initial tests of the adaptive alias procedure, applied directly to the Hasegawa–Wakatani density variable N_p (relative to which the flux flows up hill over a significant portion of the domain) and using the Padé Approximant Eq. (23) for the flux. We used Eq. (A.1) for the diffusion coefficient mapping. These revealed that the procedure yields converged solutions over very wide, but not unbounded, ranges of $D_{\rm min}$ and $D_{\rm max}$, with little difference in the convergence rate. Interestingly, this was true even for $D_{\rm min} = D_{\rm max}$, i.e., mapping onto a constant diffusion coefficient. This experience carried over to running with fluxes from the non-local turbulence code HAWCX, though with the turbulence code there may be some moderate improvement in convergence when the amount of alias redefinition required is reduced: our results for the hyperbolic-tangent test problem, Eq. (38) with $N_1 = 0.8$, appeared better converged when we took as an initial guess $n_{\rm al} = N_b$ rather than $n_{\rm al} = 0$. In Fig. 14 we show a comparison of the adaptive-alias coupled solution for $\langle N_p \rangle$ with that obtained from direct solution, averaged over 1000 iterations, of the Hasegawa–Wakatani equations, for the hyperbolic-tangent background model with $N_1 = 0.8$. The other parameters are: the window size $N = 400$ – see Eqs. (11) and (12), $D_{\rm max} = 2$, $D_{\rm min} = 1 \times 10^{-14}$, $L_x = L_y = 80\pi$, Δ_N (the scale length for variation of the background density in Eq. (38)) $= 20$, and $\alpha(x) = \alpha_0/N_b$ with $\alpha_0 = 0.000025$. As noted in Section 3.2, this test problem displays significantly non-local behavior, including flux running up the gradient of $n_{\rm tot}$ in some places. Yet Fig. 14 demonstrates that the adaptive alias procedure yields a reasonably converged density profile by 2500 iterations, which agrees reasonably well with the stand-alone solution after about 30,000 time steps; these results, and the time savings, are comparable to those achieved with the convective–diffusive split technique discussed in the text. Again we remark that a larger savings could have been demonstrated by picking a smaller κ, but then the turbulent eddies would have been a smaller fraction of the gradient scale length and thus the simulation less challenging from the standpoint of non-local coupling. Fig. 15 shows the shape of the final alias density $n_{\rm al}$, which in this case is similar to the initial guess $n_{\rm al} = N_b$. For a smart choice of initial alias, the alias is typically little modified by the adaptive alias procedure. For other choices, it can be substantially modified.

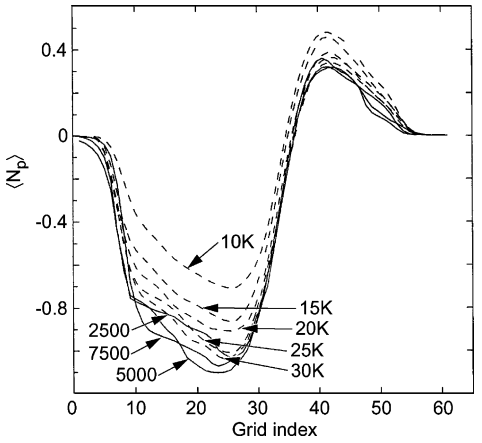

Fig. 14. Comparison of the adaptive-alias coupled solution (solid curves) for $\langle N_p \rangle$ with the stand-alone solution (dashed curves), for the hyperbolic-tangent background with $N_1 = 0.8$. The curves are plotted vs. x-grid index.

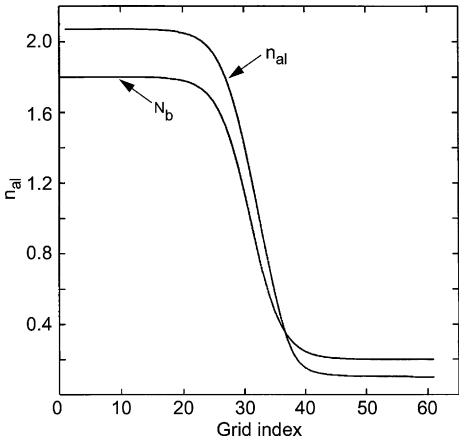

Fig. 15. The initial guess $n_{\mathrm{al}} = N_b$ and the final alias density n_{al} for the run illustrated in Fig. 14, plotted vs. x-grid index.

The final aliases corresponding to different initial choices can look completely different; but they yield the same physical density, within statistical limits, assuming both cases converge.

We have also applied the adaptive-alias procedure with $\langle N \rangle$ rather than n used as the dependent variable in the transport equation; for our strongly non-local test problem, there was little difference in performance.

References

[1] See, for example W. Horton, Drift waves and transport, Rev. Mod. Phys. 71 (1999) 735;
A.M. Dimits et al., Comparisons and physics basis of tokamak transport models and turbulence simulations, Phys. Plasmas 7 (2000) 969, and references therein.

[2] M. Kotschenreuther, W. Dorland, M.A. Beer, G.W. Hammett, Quantitative predictions of tokamak energy confinement from first-principles simulations with kinetic effects, Phys. Plasmas 2 (1995) 2381.

[3] X.Q. Xu et al., Turbulence simulations of X-point effects on the L–H transitions, in: Proc. 18th Int. Conf. on Plasma Physics and Cont. Fusion Res. Sorrento, 2000, IAEA, Vienna, 2001, IAEA-CN-77/TH4/04.

[4] A. Thyagaraja, M. de Baar, M. Romanelli, Computational physics of internal transport barriers, in: Proc. 17th Int. Conf. on the Num. Sim. of Plasmas, Banff, vol. 159, 2000.
[5] K.W. Gentle et al., An experimental counter-example to the local transport paradigm, Phys. Plasmas 2 (1995) 2292;
V.V. Parail et al., Numerical analysis of the heat pulses in JET; and P. Galli et al., Non-local response of JET H-mode discharges to temperature perturbations induced by impurity injection, in: 23rd European Phys. Soc. Conf. on Cont. Fus. and Plasma Phys., vol. 20C, Part I, The European Physical Society, Paris, 1996, pp. 35–135.
[6] M. Wakatani, A. Hasegawa, A collisional drift wave description of plasma edge turbulence, Phys. Fluids 27 (1984) 611.
[7] A.E. Koniges, J.A. Crotinger, P.H. Diamond, Structure formation and transport in dissipative drift-wave turbulence, Phys. Fluids B 4 (1992) 2785.
[8] X.Q. Xu, R.H. Cohen, J.A. Crotinger, A.I. Shestakov, Fluid simulations of non-local dissipative drift-wave turbulence, Phys. Plasmas 2 (1995) 686.
[9] I.G. Kevrekidis, C.W.J. Gear, M. Hyman, P.G. Kevrekidis, O. Runburg, K. Theodoropoulos, Equation-free multiscale computation: enabling microscopic simulators to perform system-level tasks, Commun. Math. Sci., 2002 (submitted); see also NEC Technical Report, NECI-TR 2002-010N, August 2002; http://www.arxiv.org/ps_cache/physics/pdf/0209/0209043.pdf; C. Theodoropoulos et al., "Coarse" stability and bifurcation analysis using time-steppers: a reaction-diffusion example, Proc. Natl. Acad. Sci. USA 97 (2000) 9840.
[10] L.L. LoDestro et al., Comparison of simulations and theory of low-frequency plasma turbulence, in: Proc. 13th Int. Conf. on Plasma Physics and Cont. Fusion Res., Washington, DC, 1990, vol. 2, IAEA, Vienna, 1991.
[11] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[12] Y.B. Zel'dovich, Y.P. Raizer, Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena II, Academic Press, New York, 1966.
[13] See, for example F.Y. Gang, P.H. Diamond, J.A. Crotinger, A.E. Koniges, Statistical dynamics of dissipative drift wave turbulence, Phys. Fluids B 3 (1991) 955;
B.D. Scott, The mechanism of self-sustainment in collisional drift wave turbulence, Phys. Fluids B 4 (1992) 2468;
S.J. Camargo, D. Biskamp, B.D. Scott, Resistive drift-wave turbulence, Phys. Plasmas 2 (1995) 48.
[14] S.I. Braginskii, in: M.A. Leontovich (Ed.), Reviews of Plasma Physics, vol. 1, Consultants Bureau, New York, 1965.
[15] M.A. Beer, Gyrofluid models of turbulent transport in tokamaks, Ph.D. thesis, Princeton University, January 1995.
[16] J.A. Crotinger et al., CORSICA: a comprehensive simulation of toroidal magnetic-fusion devices, Final Report to the LDRD Program, Lawrence Livermore National Laboratory Technical Report UCRL-ID-126284, March 21, 1997.