A NUMERICAL INVESTIGATION OF COLLISIONALITY AND TURBULENT TRANSPORT

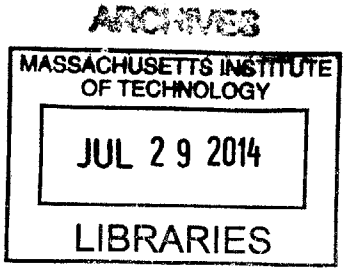
By

Martin L. Lindsey

SUBMITTED TO THE DEPARTMENT OF NUCLEAR SCIENCE
AND ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN NUCLEAR SCIENCE AND ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2014



Signature redacted

Signature of the Author: _____

Martin L. Lindsey
Department of Nuclear Science and Engineering
May 9, 2014

Signature redacted

Certified by: _____

Anne E. White
Norman C. Rasmussen Career Development Professor in Nuclear Engineering
Thesis Supervisor

Signature redacted

Accepted by: _____

Richard K. Lester
Professor and Head of the Department of Nuclear Science and Engineering

A NUMERICAL INVESTIGATION OF COLLISIONALITY AND TURBULENT TRANSPORT

By

Martin L. Lindsey

Submitted to the Department of Nuclear Science and Engineering on May 9, 2014
In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Nuclear Science and Engineering

Abstract

An investigation of collisionality's role in turbulent transport in magnetized plasma using the GS2 gyrokinetic simulation software is presented. The investigation consists of three parts, conducted by way of numerical modeling: 1) input calibration using the conditions and results of a reference investigation of a different parameter's influence on turbulence, 2) direct variation of electron-electron and ion-ion collisionality parameters, and 3) comparison between results calculated with the inclusion and exclusion of an additional collisional heating term. The calibration exercise demonstrates reliable agreement between results obtained in the present investigation and those obtained in other studies, the variation of collisionality parameters suggests a stronger dependence of ITG-driven turbulence on electron-electron collisionality than on ion-ion collisionality, and the evaluation of the collisional heating diagnostic shows a diminished influence of collisional heat drive on turbulent transport as this parameter increases. Several significant changes in some steady-state turbulent fluxes are observed at certain "threshold" values of electron-electron or ion-ion collisionality (e.g. time-averaged particle flux changing sign twice as the normalized electron-electron collisionality parameter varies between 0 and 2.5) as well as a lack of correspondence between steady-state heat, momentum and particle flux changes. These seemingly unrelated sensitivities to different ranges of collisionality parameters suggest different drives for these different transport quantities, implying a complex relationship between collisionality and turbulent heat, momentum, and particle transport of which a deeper understanding is fundamental to the design and performance of magnetic fusion projects.

Thesis Supervisor: Anne E. White
Title: Norman C. Rasmussen Career Development Professor in Nuclear Engineering

Contents

List of Figures

List of Tables

1 Introduction & background

1.1 Motivation

The advent of petascale computing systems has enabled the development and implementation of numerical physics modeling software that has otherwise exceeded computational capabilities. This direct implementation of various physics modelsm ranging from fluid dynamics to metamaterials, enables these models to be tested and explored in a domain complimentary to both theory and experiment. In particular, the problem of turbulent transport in magnetized plasma as predicted by gyrokinetic theory is the numerically modeled phenomenon that will be discussed.

1.2 Overview

Nuclear fusion & the transport problem

Magnetic confinement fusion—in which a plasma is contained by way of magnetic fields to sustain highly energetic nuclear reactions—exhibits the potential to provide abundant, affordable energy at negligible environmental cost. However, attempting to confine a plasma magnetically gives rise to an large number of instabilities, complex nonlinear phenomena, and even disruptions that can damage machinery. The goal is to maintain a plasma at high enough temperature (T) and density (n) for a sufficient length of time (τ) for it to become self-sustaining [9]. Progress toward this goal can be approximated via the *Lawson criterion*:

$$nT\tau \approx 3 \times 10^{21} (\mathrm{m}^{-3}\ \mathrm{keV\ s}). \tag{1}$$

If it is assumed that T is within 10–20 keV, with densities attainable with current magnetic fusion technology[1] of 10^{19}–10^{20} m^{-3}, the necessary τ for ignition is 1.5–3 s, which is orders of magnitude larger than the timescales over which the physics of the magnetized plasma system (associated with motion at frequencies up to $\sim 10^8$ radians/s) occur. There remains a great deal to be understood about the turbulent heat, momentum, and particle transport that occurs over these short timescales. Nonlinear gyrokinetic codes, which effectively reduce a 6D problem to 5D, can provide predictions of turbulent transport properties over a wide range of conditions that can be compared to results obtained through experiment. Codes like these have been used to model responses to changes in properties that influence and drive turbulence. The ultimate test of these tools is their ability to examine or confirm theoretical predictions (such as the suppression of turbulent fluxes by the action of sheared $\mathbf{E}\times\mathbf{B}$ flows [1] and the creation of internal transport barriers in tokamaks with negative central shear [2]). It is necessary to find the instances in which numerical models are able to make predictions that may be experimentally verified, or vice versa.

Gyrokinetic transport theory

The present transport model is gyrokinetic theory, a description of magnetized plasma that takes advantage of the disparate spatial and temporal scales governing many of its dynamics to provide approximate descriptions of low-frequency turbulence [9]. It assumes the existence of a macroscopic scale length L, which describes the distance over which equilibrium quantities vary, as well as a microscopic scale length ρ, which is the ion Larmor radius. These quantities define a fundamental expansion parameter ϵ that must satisfy the following condition:

$$\epsilon \equiv \frac{\rho}{L} \ll 1. \tag{2}$$

[1] Circa 2009.

Time scales, on the other hand, are defined in terms of three quantities. First is the ion cyclotron frequency,

$$\Omega_0 = \frac{qB}{m_i}, \tag{3}$$

where q is the ion charge, B_0 is the equilibrium magnetic field, and m_i is the ion mass. Second is the medium turbulent frequency,

$$\omega = \frac{c_s}{L}, \tag{4}$$

where c_s is the ion thermal speed (taken to be $\sqrt{T_i/m_i}$ where T_i is the ion temperature). Last is a slow transport time τ that has the following relationship to the other frequency (read: time) scales:

$$\tau^{-1} \sim \mathcal{O}(\epsilon^2\omega) \sim \mathcal{O}(\epsilon^3\Omega_0). \tag{5}$$

With these separate scales defined, it is possible to write an expression for the distribution function of position and time $f(\mathbf{r}, t)$ that contains all useful information about the magnetized plasma system in terms of several time and space variables that are defined to be mutually independent (i.e. $f(\mathbf{r}, t) \to f(\mathbf{r}_s, \mathbf{r}, t_s, t)$, where $t_s \equiv \epsilon^2 t$ is "slow time" and $\mathbf{r}_s \equiv \epsilon\mathbf{r}$ is "slow space"). The "fast" variables are then neglected when determining collective behavior. This distribution function, the electric field, and the magnetic field are then expanded in powers of ϵ—

$$\begin{aligned} f &= F_0 + \delta f_1 + \delta f_2 + \ \dots\ , \\ \mathbf{B} &= \mathbf{B}_0 + \delta\mathbf{B}, \text{ and} \\ \mathbf{E} &= \delta\mathbf{E}, \end{aligned} \tag{6}$$

where F_0 and $\mathbf{B}_0$ are the equilibrium distribution function and magnetic field respectively, $\delta f_n/F_0 \sim \mathcal{O}(\epsilon^n)$, and $|\delta\mathbf{B}|/|\mathbf{B}_0| \sim |\delta\mathbf{E}|/|c_s\mathbf{B}_0| \sim \mathcal{O}(\epsilon)$. The next assumption is that the equilibrium quantities F_0 and $\mathbf{B}_0$ only depend on the slow variables (t_s & $\mathbf{r}_s$) and the particle velocity $\mathbf{v}$, while the perturbed quantities $\delta\mathbf{B}$, $\delta\mathbf{E}$, and δf_n depend additionally on the fast variables t and $\mathbf{r}$. It should also be noted that microscopic spatial variation is assumed to only occur in the direction perpendicular to $\mathbf{B}_0$. Table 1 summarizes these dependencies.

Table 1: Dependencies of plasma quantities on slow and fast variables.

Function	Dependence
F_0	$\mathbf{r}_s$, $\mathbf{v}$, t_s
δf	$\mathbf{r}_\perp$, $\mathbf{r}_s$, $\mathbf{v}$, t, t_s
$\mathbf{B}_0$	$\mathbf{r}_s$, $\mathbf{v}$, t_s
$\delta\mathbf{B}$	$\mathbf{r}_\perp$, $\mathbf{r}_s$, $\mathbf{v}$, t, t_s
$\delta\mathbf{E}$	$\mathbf{r}_\perp$, $\mathbf{r}_s$, $\mathbf{v}$, t, t_s

These assumptions result in the *gyrokinetic ordering*, which states that the space and time derivatives of the equilibrium quantities are an order smaller than the space and time derivatives of the perturbed quantities. This is expressed by the list of statements below:

- $\nabla \sim \mathcal{O}(L^{-1})$ [acting on F_0 or $\mathbf{B}_0$]

- $\frac{\partial}{\partial t} \sim \mathcal{O}(\epsilon^2 \omega)$ [acting on F_0 or $\mathbf{B}_0$]
- $\nabla_\perp \sim \mathcal{O}(\rho^{-1})$ [acting on δf_1, $\delta \mathbf{B}$, or $\delta \mathbf{E}$]
- $\nabla_\parallel \sim \mathcal{O}(L^{-1})$ [acting on δf_1, $\delta \mathbf{B}$, or $\delta \mathbf{E}$]
- $\frac{\partial}{\partial t} \sim \mathcal{O}(\omega)$ [acting on δf_1, $\delta \mathbf{B}$, or $\delta \mathbf{E}$]

This separation of space and time scales enables such an expansion of the distribution function and electromagnetic fields, a critical step in solving the Fokker-Planck equation (expressed here in terms ordered in powers of ϵ:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = C(f, f), \tag{7}$$

where $C(f, f)$ describes the effects of collisions on the distribution function. It is this equation, when solved at successive orders of ϵ, that leads to the numerical systems typically implemented in turbulence codes.

1.3 Objectives

Parameter calibration: E×B flow shear and turbulence

A set of "initialization runs" of the code replicating an investigation of the effects of $\mathbf{E}\times\mathbf{B}$ flow shear on turbulence can verify possible differences between the runs of GS2 to follow and other results obtained with nonlinear gyrokinetics codes [3]. In this sense, these relative differences between the respective base cases of each simulation can be inferred.

Dependence of turbulence on intra-species collisionality

The next objective is to use GS2 to conduct an investigation of the effects of collisionality—a dimensionless figure of merit comparing the timescale of its particles thermal transits to that of its collision times—on turbulent transport in tokamak plasmas. The collisionality parameter was varied over three orders of magnitude with the intention of observing the resulting effects on turbulent fluxes. The focus on collisionality in particular is motivated by results from collisionality changes in the Alcator C-Mod tokamak that changes in momentum transport did not correlate with changes in particle transport [4]. The GYRO code failed to reproduce the observed momentum transport properties, so a systematic next step would be to use an alternative tool, GS2, to simulate the conditions of the experiment.

The collisional heating diagnostic

Despite how computationally intensive nonlinear gyrokinetic codes are, many aspects of the physics are simplified or modeled only approximately for optimal numerical implementation and performance. In particular, the problem of modeling the effects of small angle Coulomb collisions on an arbitrary distribution (i.e. numerically implementing the linearized Landau operator [5]) lies beyond current limits on computational resources. The various strategies employed to model energy diffusion in phase space with a collision operator have varying degrees of reliability, and one strength of GS2's collision operator is that it maintains the H-Theorem ($\frac{dS}{dt} \geq 0$, i.e. entropy cannot decrease) and conservation laws. Since it is noted in the explanation of the collision operator that all irreversible heating is ultimately collisional, the collisional heating diagnostic

(though switched off by default) would presumably provide information on the amount of heat generated via turbulent dissapation. A comparison between having the diagnostic switched on and having it switched off also allows the performance associated with the diagnostic to be evaluated.

2 Tools & methods

2.1 The Gyrokinetic Equation

Many software packages[2] developed to model gyrokinetic turbulence make use of *gyro-center variables* defined below—

$$\begin{aligned} \mathbf{R} &\equiv \mathbf{r} + \frac{\mathbf{v} \times \mathbf{b}_0}{\Omega_0}, \text{ (guiding center position)} \\ \varepsilon &\equiv \frac{1}{2} m v^2 + q\varphi, \text{ (particle energy) and} \\ \mu &\equiv \frac{m v_\perp^2}{2B_0}, \text{ (magnetic moment)} \end{aligned} \tag{8}$$

where $\mathbf{b}_0$ is the direction of the equilibrium magnetic field, φ is the electrostatic potential, and $v_\perp$ is the velocity perpendicular to the equilibrium magnetic field. With these variables, the distribution function becomes $f = F_0(\mathbf{R}, \varepsilon, t) + \delta f_1(\mathbf{R}, \varepsilon, \mu, t) + \delta f_2(\mathbf{R}, \varepsilon, \mu, \vartheta, t) + \ \ldots$, where ϑ is the gyroangle. Then, (7) can be expressed as

$$\frac{\partial f}{\partial t} + \frac{d\mathbf{R}}{dt} \cdot \frac{\partial f}{\partial \mathbf{R}} + \frac{d\mu}{dt}\frac{\partial f}{\partial \mu} + \frac{d\varepsilon}{dt}\frac{\partial f}{\partial \varepsilon} + \frac{d\vartheta}{dt}\frac{\partial f}{\partial \vartheta} = C(f, f), \tag{9}$$

(where $C(f, f)$ describes the physics of collisions) which can be *gyro-averaged*, a process defined below for an arbitrary function X—

$$\langle X(\mathbf{r}, \mathbf{v}, t) \rangle_{\mathbf{R}} \equiv \frac{1}{2\pi} \int_0^{2\pi} X(\mathbf{R} - \frac{\mathbf{v} \times \mathbf{b}_0}{\Omega_0}, \mathbf{v}, t) d\vartheta. \tag{10}$$

The gyroaverage is an average over ϑ at fixed gyrocenter $\mathbf{R}$—it grants a description of the collective behavior of quantities that occurs over many particle orbits[3]. With the additional definitions of

$$\begin{aligned} \chi &\equiv \varphi - \mathbf{v} \cdot \delta\mathbf{A} \text{ and} \\ \delta f_1(\mathbf{r}, \mathbf{v}, t) &\equiv \delta f_{1,cl}(\mathbf{R}, v, v_\perp, t) + h(\mathbf{R}, v, v_\perp, t), \end{aligned} \tag{11}$$

where $\mathbf{A}$ is the magnetic vector potential, $\delta f_{1,cl}$ is the classical part of the first-order component of f, and h is the gyrokinetic part of of the first order component of f (usually called the *guiding center distribution*). It is possible to then arrive at the Gyrokinetic Equation governing the dynamics of h:

$$\frac{\partial h}{\partial t} + (v_\parallel \mathbf{b}_0 + \mathbf{v}_D + \mathbf{v}_\chi) \cdot \nabla h - \langle C[h] \rangle_{\mathbf{R}} = q\frac{F_0}{T_0}\frac{\partial \langle \chi \rangle_{\mathbf{R}}}{\partial t} - \mathbf{v}_\chi \cdot \nabla F_0, \tag{12}$$

where $v_\parallel$ is the velocity parallel to the equilibrium magnetic field, $\mathbf{v}_D$ is the equilibrium drift velocity, $\mathbf{v}_\chi$ is the perturbed drift velocity, and $C[h]$ is the *collision operator* that represents the

[2]Such as GENE, GYRO, GTK, GTS, GTC, GEM, M3D, AstroGK, etc.

[3]For example, the gyro-average of the time-derivative of $\mathbf{R}$, $\langle \frac{d\mathbf{R}}{dt} \rangle_{\mathbf{R}}$, provides an expression for the drift velocity of the particle's gyrocenter.

effects of collisions on h. The Gyrokinetic Equation can be thought of as describing the time evolution of rings of charge centered at guiding center position $\mathbf{R}$ subject to spatially variating fields [7]. Nonlinear turbulence codes have different ways of implementing and solving this equation numerically.

2.2 The GS2 software package

As with any numerical investigation of physics, it is important to analyze the details of the calculations being performed by the code. Such an analysis can point out possible strengths or weaknesses of the code in predicting its output physical quantities and reveal sources of error. GS2 is a continuum flux tube code capable of simulating toroidal geometry, general axisymmetric plasma shapes[4], multiple species, trapped and passing non-adiabatic electrons, electromagnetic fluctuations, collision operators, and equilibrium scale $\mathbf{E}\times\mathbf{B}$ shear flow [7].
The output quantities calculated by GS2 that will be studied below, along with their definitions (in which the overline represents a spatial average) are:

- Heat flux:
$$Q_s = \overline{\int d^3\mathbf{v} \frac{m_s v^2}{2} \mathbf{v}_\chi h_s} \tag{13}$$
- Momentum flux:
$$\Pi = \overline{\sum_s m_s R^2 \int d^3\mathbf{v} (\mathbf{v}\cdot\nabla\phi)\mathbf{v}_\chi h_s} \tag{14}$$
- Particle flux:
$$\Gamma_s = \overline{\int d^3\mathbf{v}\, \mathbf{v}_\chi h_s} \tag{15}$$
- Energy exchange:
$$\Delta E_s = \overline{\int d^3\mathbf{v} \frac{1}{2} m_s v^2 C(f_s, f_{s'})} \tag{16}$$

where ϕ is the toroidal angle, R is the major radius of the torus, and the subscripts denote that quantities correspond to species s or s'.

2.3 Simulations in GS2

GS2 is capable of fully gyrokinetic, nonlinear (as well as linear) simulations with good scaling to many processors [8]. It takes a collection of input values, performs a set of calculations within the gyrokinetic approximation, and periodically records the output values as the calculations progress. The following sections concern some of the language that will be used to discuss the simulations, the numerical factors used by GS2 to scale each physical quantity, and some typical input parameters used in studies of turbulent transport.

2.3.1 Terminology

After an input file (an example of which is contained in appendix D) containing the necessary settings is provided, a *run*, or series of calculations based on a particular set of input conditions, may be performed. Resulting values are recorded in an output file at user-specified intervals for a user-specified number of time-steps. Here, a *trial* refers to a series of runs over which a chosen input parameter is varied while all other inputs remain unchanged.

[4]Even stellerators.

2.3.2 Normalizations & units

The quantities represented in GS2 are *normalized*, or multiplied by certain reference units, to obtain unitless values for optimal calculation. Thus, there are conversion factors between values contained in the output file and the corresponding physical quantities they represent. The reference units used for normalizations are defined in Table 2. In general, units' definitions depend on the choice of geometry—as will be discussed, the geometry of interest is the `Cyclone` base case, leading to some simplifications[5].

Table 2: Factors used to normalize GS2 quantities.

Factor	Definition	Description
a	-	half the diameter of the last closed flux surface (i.e. minor radius)
r	-	radial coordinate of flux surface (i.e. $0 \leq r \leq a$)
R_a	-	major radius (in the "circular flux surface" case)
Ψ	-	magnetic flux; used to define magnetic flux density (i.e. field)
T_i, p_i, n_i	-	temperature, pressure, & density of ion species
m_i	-	ion mass
ρ_a	r/a	flux surface label*
$v_{th,i} = c_s$	$\sqrt{2T_i/m_i}$	ion thermal velocity
B_a	$I(\Psi)/R_a$	toroidal magnetic field evaluated at a given flux surface
Ω_a	$\lvert e\rvert B_a/m_i c$	gyrofrequency of a species at a given flux surface
$\rho_{i,a}$	$v_{th,i}/\Omega_a$	gyroradius* of ion species at a given flux surface

*Care must be taken so that $\rho_{i,a}$ (ion gyroradius) is not confused with ρ_a (flux surface label).

The normalizations (in terms of the factors defined in Table 2) for the output quantities of interest are shown in Table 3.

Table 3: Normalizations for quantities output by GS2.

Quantity	Normalization
t	c_s/a
Π_s	$a/c_s^2\rho_{i,a}^2 m_i n_i$
Q_s, $(\Delta E)_s$	$a^2/p_i c_s \rho_{i,a}^2$
γ_E	a/c_s
B	$1/B_a$
φ	$\lvert e\rvert a/T_i \rho_{i,a}$

2.3.3 Core inputs

The input parameters are stored in a file that consists of many *namelists*, or lists of input values used by its corresponding module [13]. It is useful to adopt a consistent set of code parameters, or a set of "core inputs" that allow for meaningful comparisons to be made between series of runs. A

[5]Other cases are beyond the present scope; more information can be found in the GS2 wiki.

distinction will be made between the inputs specifying the geometry of the system, the remaining physical inputs, and computational inputs.
The `Cyclone` base case of an unshifted, circular flux surface is a standard magnetic geometry commonly used to study ITG (Ion Temperature Gradient)-driven turbulence [1]. Its properties are listed in Table 4[6].

Table 4: `Cyclone` base case parameters [11].

Parameter	Description	Input	Value
R/L_{Ti}	normalized inverse ion temperature gradient length scale	`tprim`	6.9
R/L_n	normalized inverse electron density gradient length scale	`tprim`	2.2
q	magnetic safety factor	`qinp`	1.4
$\hat{s}$	magnetic shear	`shat`	0.8
T_i/T_e	ion-electron temperature ratio	`tite`	1.0
Z_{eff}	effective ionic charge	`zeff`	1.0
β	ratio of ion pressure to magnetic energy density	`beta`	0.0

By deviating from this base case, it is possible to investigate the consequences of microturbulence on a variety of physical quantities of interest.
The other normalized quantities that are plugged into the numerical solver to calculate resulting physical quantities are referred to as physical inputs. These inputs, when varied, allow the physics modeled by `GS2` to be probed. Table 5 describes the most relevant physical inputs.

Table 5: Core physical inputs for each species used in each run. These numerical values are normalized in terms of the ions (i.e. the electron mass is expressed in terms of ion mass).

Input	Value (ions)	Value (electrons)	Description
`z`	1.0	-1.0	charge
`dens`	1.0	1.0	density
`mass`	1.0	2.7×10^{-4}	mass (normalized to ions)
`temp`	1.0	1.0	temperature (normalized to ions)
`vnewk`	7.07107×10^{-4}	7.07107×10^{-4}	intra-species collisionality parameter
`g_exb`	0.0	0.0	E×B flow shearing rate

The remaining inputs of interest concern the specifics of *how* `GS2` computes its results. They determine, for example, how often to record its output quantities or whether to save its progress for restarting in another run; they ultimately affect the resolution, performance, and error accumulation of the code. In fact, the distinction between `GS2` performing linear and nonlinear calculations, as well as that between it treating electrons adiabatically or kinetically in its calculations both depend on computational inputs. The most relevant of these inputs are contained in Table 6.

2.4 Analysis of outputs: the `Julia` programming language

`GS2` records its outputs in the NetCDF files, a commonly used format in scientific computing. There are countless programming languages capable of processing and manipulating NetCDF

[6]It should be noted that this case is also defined as treating the electrons as adiabatic, but that specification was only met for one of the trials presented here.

Table 6: Core computational inputs used in each run. These inputs are designed so that each run takes ~12 hours of real time on a petascale high-performance computing system.

Input	Value	Description
nstep	50000	number of time-steps to advance
nwrite	50	number of time-steps between output recordings
ny	32	number of k_y modes to include in calculations
nx	128	number of k_x modes to include in calculations
vcut	2.5	number of standard deviations from the standard Maxwellian beyond which F_0 is zeroed
nspec	2	number of species to include (set to 1 to model electrons adiabatically)
collision_model	'default'	determines which collision model to use ('default' includes pitch angle scattering and energy diffusion)
heating	.false.	toggles collisional heating diagnostic (see §3.3)
ginit_option	'noise'	sets the way that the distribution function is initialized ('noise' is the recommended default, and 'many' is used to restart a run)
nonlinear_mode	'on'	toggles between linear and nonlinear mode

data in particular, and the choice of analysis software is largely a matter of personal preference. The relatively new Julia language, developed in 2012 to combine the performance of traditional scientific computing languages (e.g. C & Fortran) with the usability of high-level dynamic languages (e.g. Python & MATLAB) is used in this case. The automated script (contained in appendix D) that prepares a computing environment for the analysis and plotting of GS2 data is written in this language for the following reasons:

- to simplify the processes of writing, editing, and understanding the code used for analysis
- to promote Julia as a free, open-source scientific computing standard
- to explore the capabilities of a relatively new programming language

3 Trials

3.1 Parameter calibration: E×B flow shear and turbulence

As a way to verify the results produced by GS2 and calibrate its operation, a series of runs corresponding to a range of E×B flow shears was executed. The goal was to recreate a certain plot that shows a number of local maxima in toroidal angular momentum flux within a certain range of γ_E, or flow shearing rate. E×B flow shear was chosen in particular because of the relatively large numer of studies that have been conducted on its effect on plasma stability as well as its common numerical implementation in simulation codes.

Context

In GS2, the rate of E×B flow shear is modeled using the (normalized) parameter g_exb. The un-normalized flow shearing rate, with units of inverse time, is defined as:

$$\gamma_E = \frac{\rho_a}{q}\frac{d\omega}{d\rho_a} \tag{17}$$

where ω is the toroidal angular velocity [13]. This quantity is then multiplied by a/c_s to obtain g_exb. In general, the expected effects of $\mathbf{E}\times\mathbf{B}$ flow shear are to produce Floquet modes, or oscillations that carry fluctuations through regions where the magnetic field line curvature alternates between stabilizing and destabilizing [1].

Setup

The Cyclone base case was used, matching the input conditions of the mimicked study. The value of g_exb was varied over a range of values corresponding to the figure from [1]. GS2 was set to model electrons adiabatically. Outputs were recorded at 1000 time-steps, and runs reached normalized times $t(c_s/a)$ in the range of 400-600. Each run was resumed to record output quantities at an additional 1000 time-steps to ensure that steady-state was achieved, and the initial startup period (explained below) was excluded in computing the time averages.

Results

Since the electrons were treated adiabatically, all electron fluxes were identically zero—the plots from this trial depict quantities calculated for ions. The ten sampled values of g_exb were 0.01, 0.033, 0.0533, 0.07, 0.1067, 0.16, 0.2133, 0.2667, 0.32, and 0.3733. Time histories of turbulent heat, momentum, and particle fluxes as well as inter-species energy exchange were obtained at each probed value of shearing rate. The full set of time trace plots is located in appendix A. An example time trace is shown in Fig. 1.

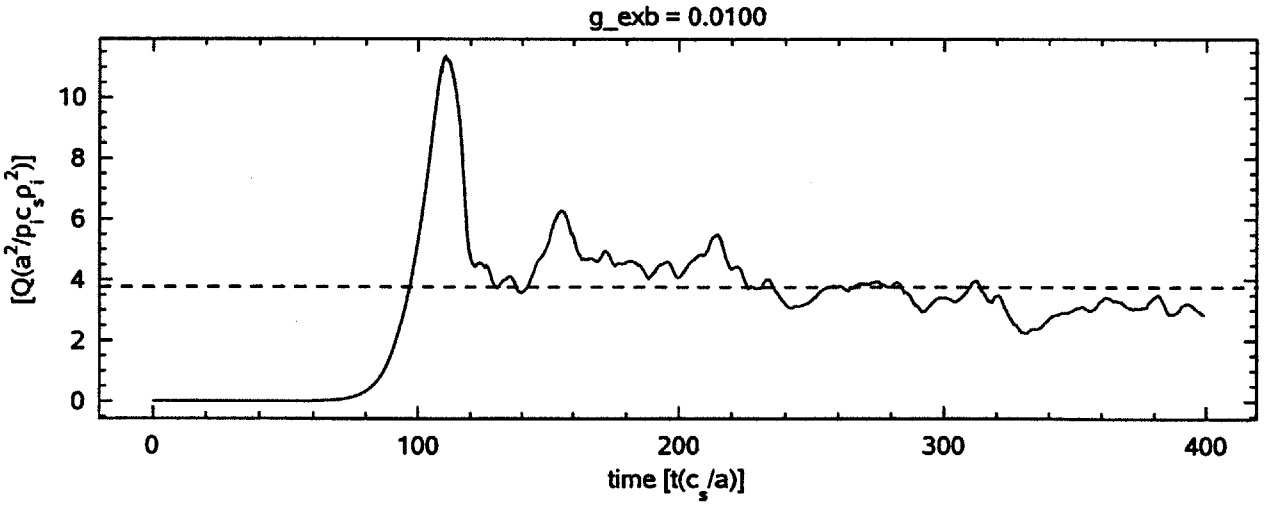


Figure 1: Example time trace of the computed ion heat flux. It exhibits the characteristic behavior of staying near zero for some time at first before the turbulence begins around $t(c_s/a) \sim 120$.

The most surprising characteristic of this trial is the sudden and significant decline in the magnitude of turbulent fluxes after a certain value of g_exb. Between 0.2133 and 0.2667, for each flux in question, its time average decreases by several orders of magnitude. In fact, at g_exb = 0.2667 and above, none of the runs converged because turbulence was so heavily suppressed.

Analysis

Time averages of all quantities were computed using the normalized time ranges specified in Table 7. In general, the linear startup phase grew longer with increasing g_exb, in agreement with the intuitive notion that flow shear acts to suppress turbulence. Consequently, at the highest values of flow shear parameter, it appears as though turbulence was suppressed so much that no quantities deviated significantly from zero, and no steady-state turbulent time-evolution can be observed. Plots of the resulting time averaged ion fluxes are contained in Fig. 2.

Table 7: Ranges over which time-averaged fluxes are computed. The g_exb = 0.2667, 0.3200, & 0.3733 runs did not converge within 50,000 time-steps and remained close to zero in value.

g_exb	Normalized time range
0.0100	139.24 – 399.24
0.0330	359.08 – 595.96
0.0533	359.23 – 595.48
0.0700	339.16 – 558.84
0.1067	399.07 – 627.83
0.1600	449.01 – 660.51
0.2133	599.13 – 756.63

For comparison, the plots from a similar investigation[7] that inspired this trial is included in Fig. 3. This preliminary test of GS2 demonstrates satisfactory agreement with results obtained in other studies—it supports the assertion that E×B flow shear is associated with the suppression of microturbulence and ultimately lead to smaller linear growth rates of instabilities. Similarly, it identifies a critical value of γ_E (between g_exb = 0.2133 & 0.2667) above which turbulence is strongly suppressed. This agreement provides, in some sense, a benchmark comparison that helps to validate further results obtained with GS2.

[7]The γ_E that can be seen in these plots differ from g_exb by a factor of $\sqrt{1/2}$.

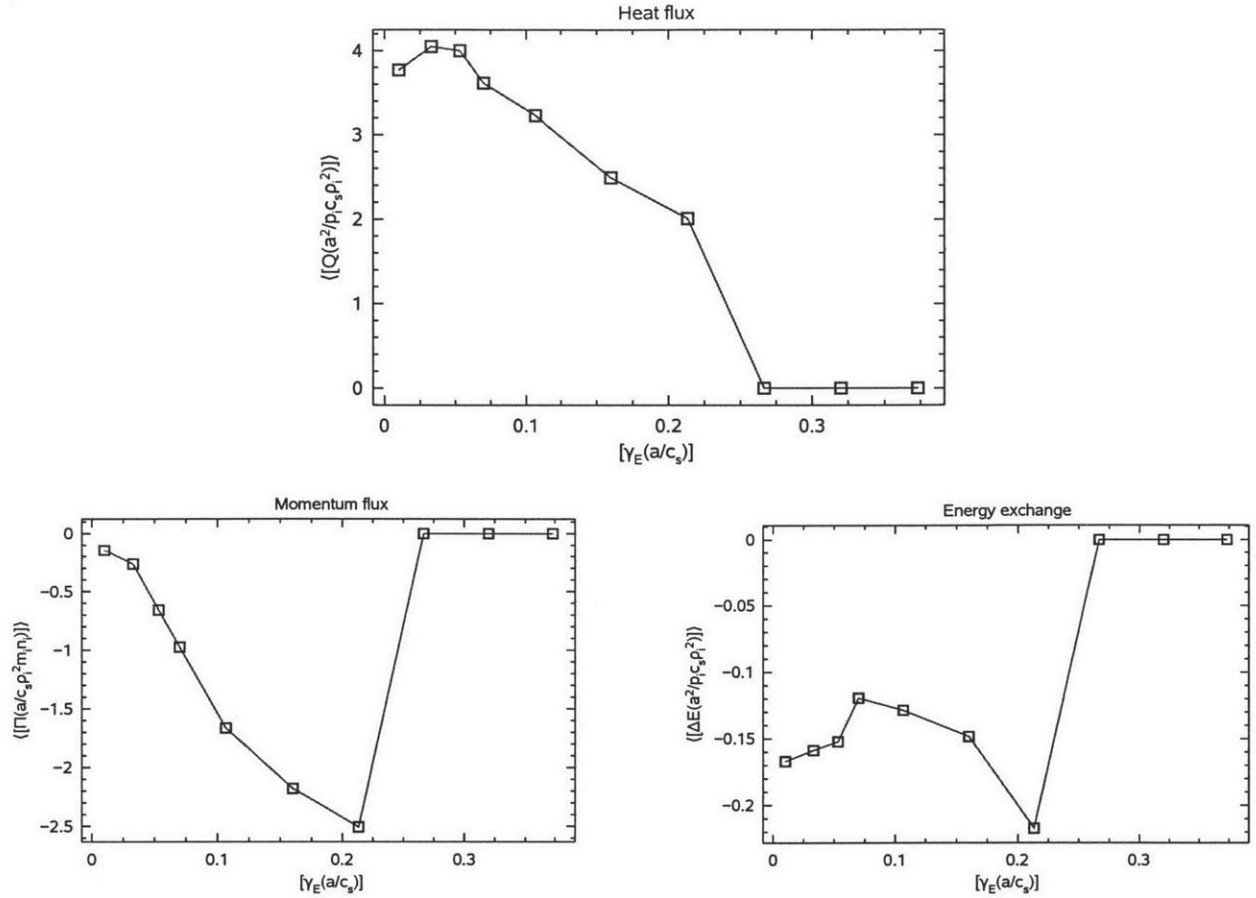


Figure 2: Variation of time-averaged ion quantities with respect to shearing rate parameter `g_exb`. Each plot shows a marked jump to zero after this parameter increases beyond 0.2133, suggesting a possible critical value of γ_E at which turbulence is significantly suppressed.

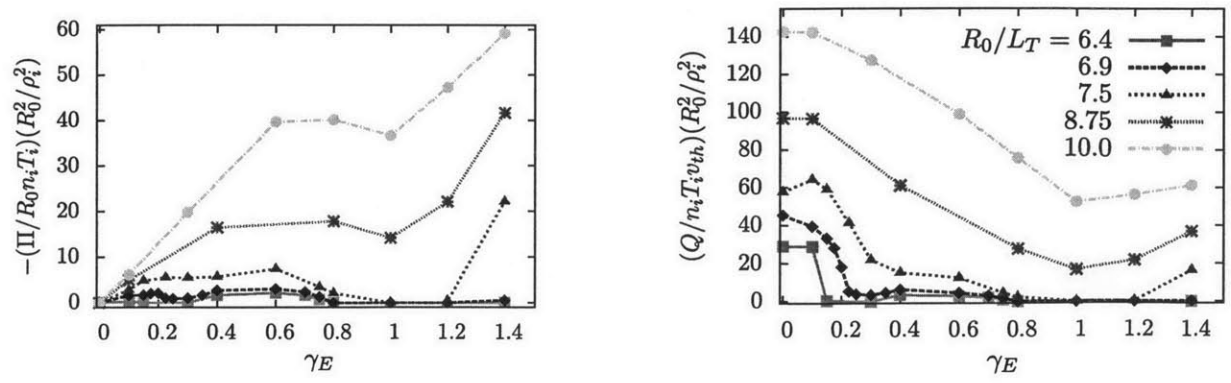


Figure 3: Reference plots from a similar `GS2` investigation [1] that shows qualitative agreement with the presently obtained results (corresponding to the $R_0/L_T = 6.9$ curve).

3.2 Dependence of turbulence on intra-species collisionality

While various gyrokinetic codes have been developed to study the effects of low-frequency microturbulence on magnetized plasmas, few chances have been taken to specifically examine its dependence on collisionality[8]. Many opportunities for comparison of gyrokinetic theory (as well as its computational counterpart) with experiment through the variation of other parameters thus remain open. The aim of the present trial was to explore how turbulent transport properties of tokamak plasmas computed with GS2 vary with respect to both electron and ion collisionality (defined below). Time averages of calculated turbulent fluxes will be examined, and their variation with respect to changes in collisionality will be discussed.

Context

Collisionality is a dimensionless figure of merit comparing the timescale of a particle's thermal transits to that of its collisions. It can be defined for intra-species collisions (i.e. ν_{ee} or ν_{ii}) as well as inter-species collisions (ν_{ie}). The parameters used within GS2 to respectively model electron and ion collisionality are defined as:

$$\nu_{ee}^{GS2} = \sqrt{\frac{2}{m_e}}\frac{a}{c_s}\frac{\pi n_e e^4 \ln(\Lambda)}{T_e^{3/2}} \quad \text{and} \quad \nu_{ii}^{GS2} = \sqrt{\frac{2}{m_i}}\frac{a}{c_s}\frac{\pi Z_i^2 Z_{\text{eff}} n_e e^4 \ln(\Lambda)}{T_i^{3/2}}, \tag{18}$$

where e is the elementary charge, $\ln(\Lambda)$ is the Coloumb logarithm, Z_s is the species' charge, T_s is its temperature, n_s is its density, and m_s is its mass [13]. GS2 is designed so that ν_{ee} and ν_{ii} can effectively be set independently for each species, but it is not currently apparent how to record or impose a value for ν_{ie}. Thus, each collisionality parameter was varied independently (i.e. first ν_{ee}, then ν_{ii}) via the process detailed below.

Setup

For each species, a series of runs were executed wherein the intra-species collisionality parameter, vnewk (defined in (18)) was varied. For both electron and ion collisionality, the six values of vnewk that were probed are 0.005, 0.05, 0.1, 0.5, 1.0, and 2.5. The electron and ion intra-species collisionality parameters used in GS2 will henceforth be referred to as $\text{vnewk}_{(ee)}$ and $\text{vnewk}_{(ii)}$, respectively. All other parameters remained unchanged between runs, and the non-scanned vnewk parameter was left at the default value of $\sqrt{2} \times 10^{-3}$ (i.e. $\text{vnewk}_{(ii)} = 0.000707107$ in the set of $\text{vnewk}_{(ee)}$ runs and vice versa). Electrons were modeled kinetically. Each run recorded outputs at 1000 timesteps and generally reached a normalized time of $t(c_s/a) = 200 - 300$. Each of the runs in the $\text{vnewk}_{(ee)}$ set were resumed to record outputs at another 1000 timesteps, eventually reaching $t(c_s/a) = 400 - 600$. Larger timesteps were associated with higher values of vnewk for both the ion and electron sets of runs.

Results

An example time trace is shown below in Fig. 4; the others can be found in appendix B.

Analysis

The time coordinates used to define the excluded portions of each run are listed below in Table 8.

[8] This is related in part to the challenge of reliably simplifying the physics of charged particle collisions for numerical implimentation.

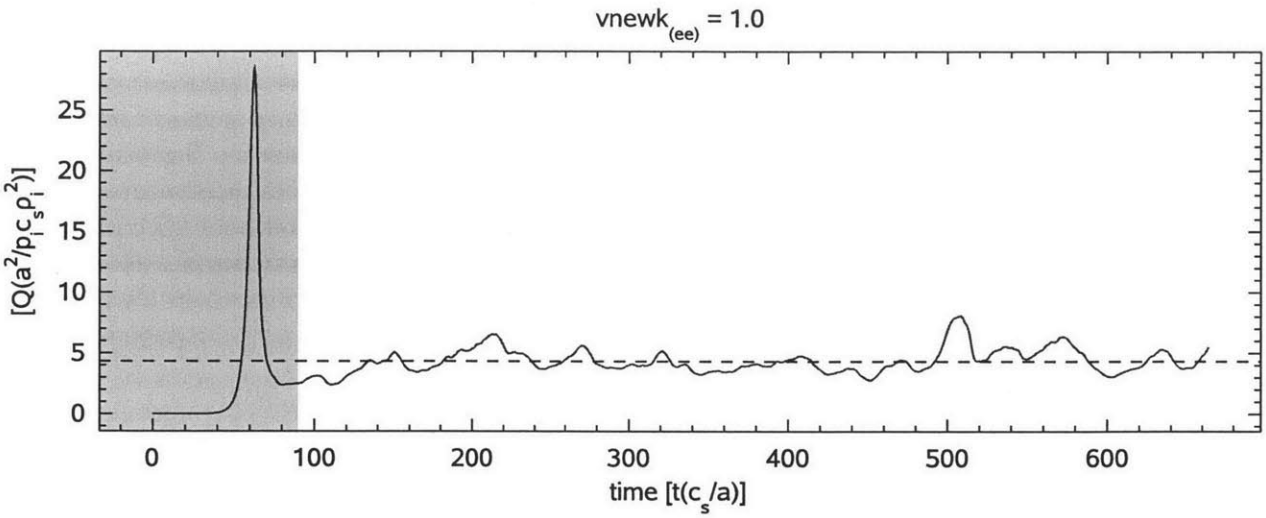


Figure 4: Example time trace of turbulent ion heat flux. The shaded region represents the portion of the run that is excluded from the calculated time-averaged values. These portions of each run (i.e. each value of vnewk) can be defined in turbulent heat flux time histories as containing the characteristic linear growth phase and are excluded from the time averages of other quantities (e.g momentum and particle flux).

Table 8: Normalized time ranges determined by turbulent heat flux time traces.

$\text{vnewk}_{(ee)}$	Normalized time range
0.005	49.03 – 347.82
0.05	59.06 – 425.16
0.1	69.12 – 617.31
0.5	79.29 – 659.14
1.0	89.25 – 664.05
2.5	99.25 – 720.74

$\text{vnewk}_{(ii)}$	Normalized time range
0.005	49.11 – 191.92
0.05	59.01 – 199.04
0.1	69.09 – 228.53
0.5	79.29 – 314.40
1.0	89.16 – 341.66
2.5	99.09 – 353.47

The variations in time averages of the normalized heat flux with respect to electron collisionality ($\text{vnewk}_{(ee)}$) is shown in Fig. 5. Figs. 6, 7 & 8 respectively contain similar plots of momentum flux, particle flux, and energy exchange.
The ion collisionality ($\text{vnewk}_{(ii)}$) dependence of time-averaged (normalized) heat flux is shown in Fig. 9. Figs. 10, 11 & 12 respectively contain similar plots of momentum flux, particle flux, and energy exchange.
Most turbulent flux averages appeared to approach small values as $\text{vnewk}_{(ee)}$ increased beyond 1. For both ions and electrons, the heat flux decreases by about a factor of two for $0 < \text{vnewk}_{(ee)} \leq 1$ before falling more slowly. The momentum flux typically remained close to zero but appears to decrease at least slightly as $\text{vnewk}_{(ee)}$ increases from zero. Particle flux is identical between ions and electrons following a pattern of going from being negative-valued to positive-valued as $\text{vnewk}_{(ee)}$ increases from 0 to 1 and back down to negative-valued as $\text{vnewk}_{(ee)}$ crosses over and exceeds 1. This is surprising, as it seems to suggest a net inward or outward flows of particles toward the core as electron collisionality changes, as well as some threshold value ν'_{ee} at which particle flux changes sign. The amount of energy exchanged between ions and electrons was always negative for the ions and positive for the electrons, but this value appeared to be greatest when $\text{vnewk}_{(ee)} \ll 1$ and decreased as $\text{vnewk}_{(ee)}$ grew larger. Additional runs at values of $\text{vnewk}_{(ee)}$

Heat flux (vnewk$_{(ee)}$ set)

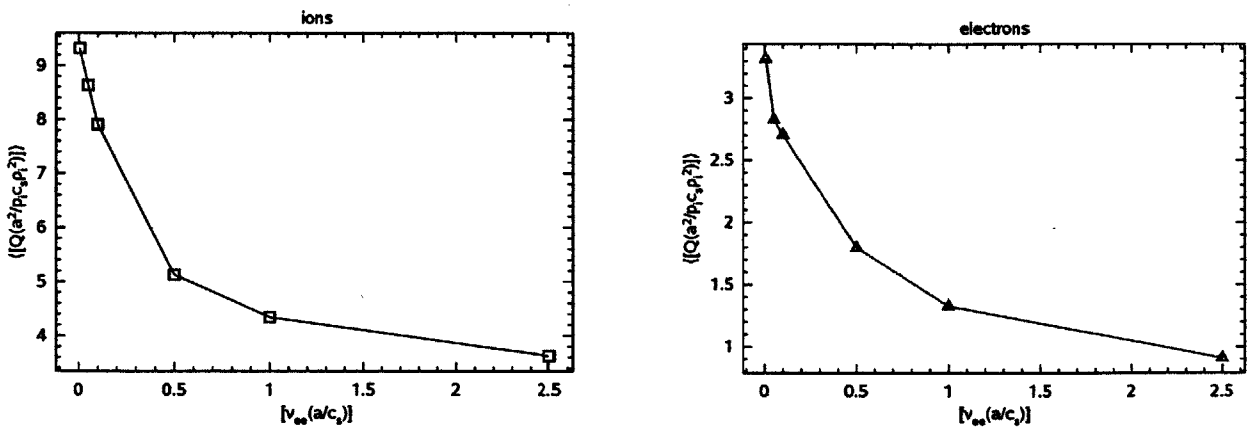


Figure 5: Variation of time-averaged heat flux with vnewk$_{(ee)}$.

Momentum flux (vnewk$_{(ee)}$ set)

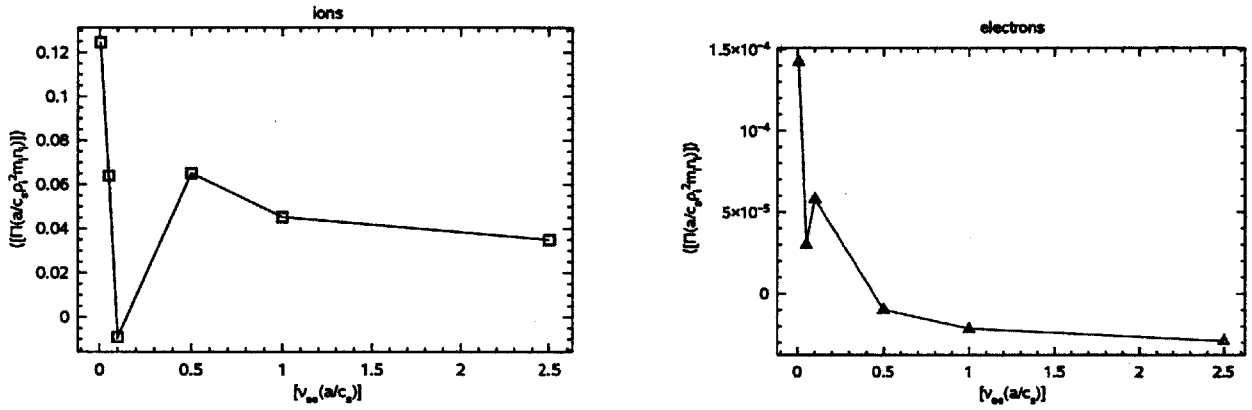


Figure 6: Variation of time-averaged momentum flux with vnewk$_{(ee)}$.

between 1 and 2 could clarify some trends in the data (or point out a lack thereof).
The scans across the chosen range of values show that estimates of time-averaged heat turbulent heat and particle fluxes tended to decrease as ν_{ii} increased for both ions and electrons. For the heat flux, this decrease is smaller in comparison to that due to a similar decrease in ν_{ee} over the same range (a factor of $\sim$1/5 vs. a factor of $\sim$2), while the particle flux changes in somewhat different ways as either ν_{ee} or ν_{ii} is scanned, taking on only negative values for each scanned vnewk$_{(ii)}$. Similarly to the results of the set of vnewk$_{(ee)}$ runs, the time average of the momentum flux appears to generally stay close to zero in absolute value, but it shows greater variation between scanned values with ν_{ii} than with ν_{ee}. The time-averages of the electron energy exchange steadily decrease as ν_{ii} increases and actually cross zero before vnewk = 1. This is very different from the corresponding energy exchange time-averages from the ν_{ee} scans—in those runs, none of the time-averaged energy exchange values were negative for electrons, and they are typically larger in magnitude. The time average of the exchanged ion energy was negative at all probed values of vnewk$_{(ii)}$, just as they were for vnewk$_{(ee)}$ runs.

Particle flux (vnewk$_{(ee)}$ set)

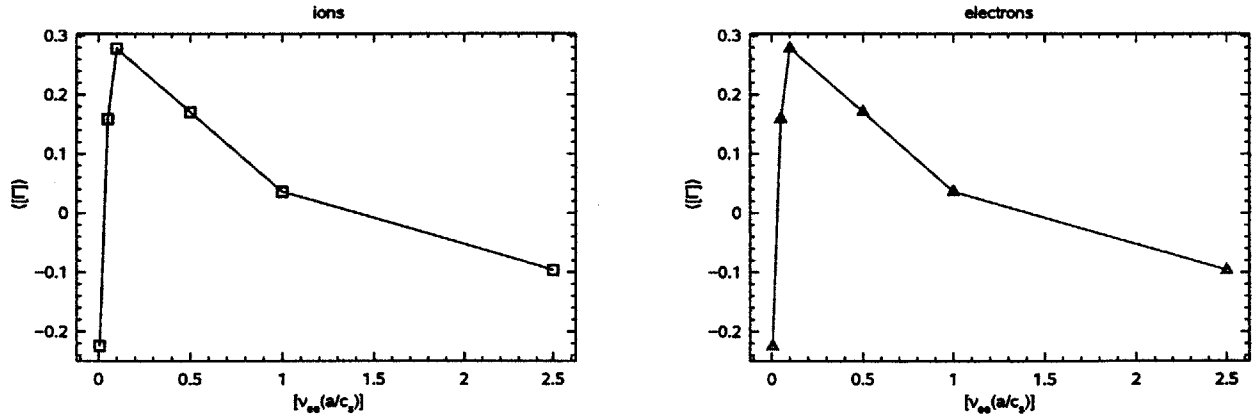


Figure 7: Variation of time-averaged particle flux with vnewk$_{(ee)}$.

Energy exchange (vnewk$_{(ee)}$ set)

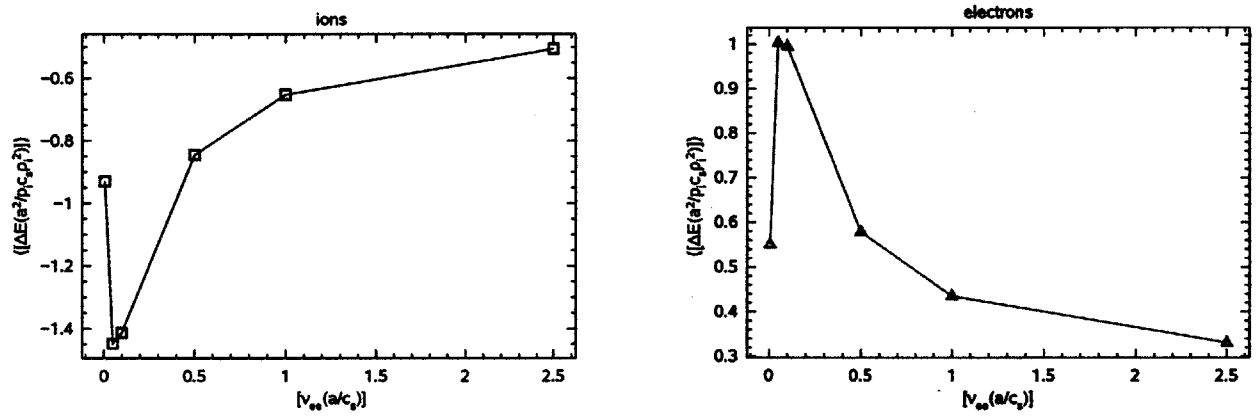


Figure 8: Variation of time-averaged energy exchange with vnewk$_{(ee)}$.

Heat flux (vnewk$_{(ii)}$ set)

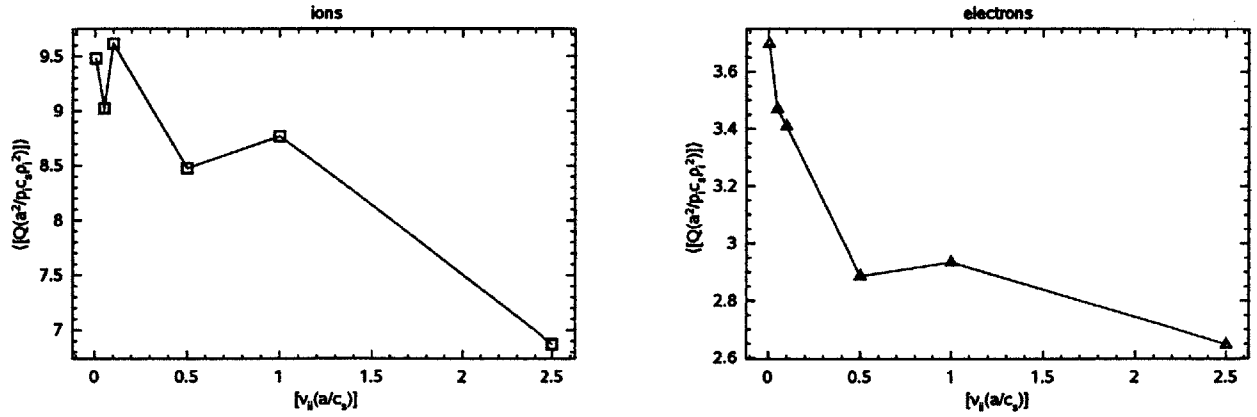


Figure 9: Variation of time-averaged heat flux with vnewk$_{(ii)}$.

Momentum flux (vnewk$_{(ii)}$ set)

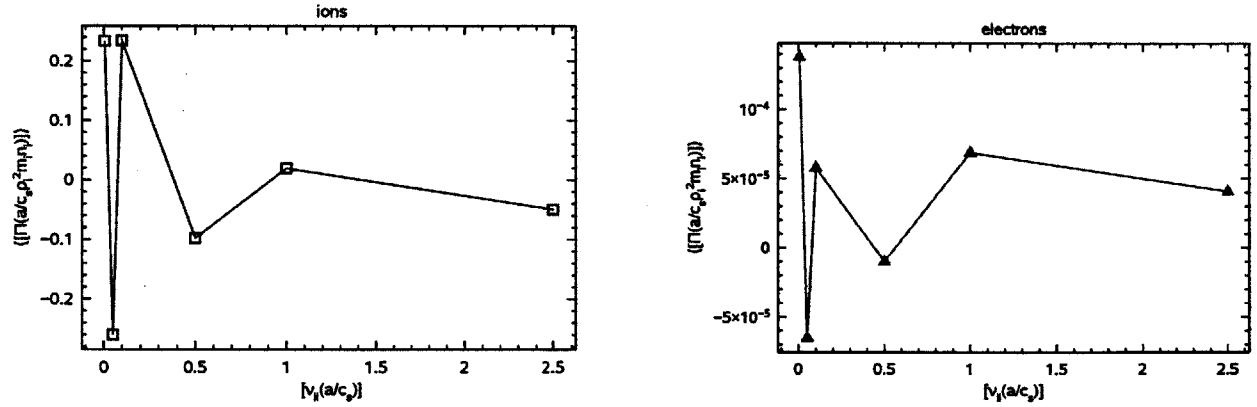


Figure 10: Variation of time-averaged momentum flux with vnewk$_{(ii)}$.

Particle flux (vnewk$_{(ii)}$ set)

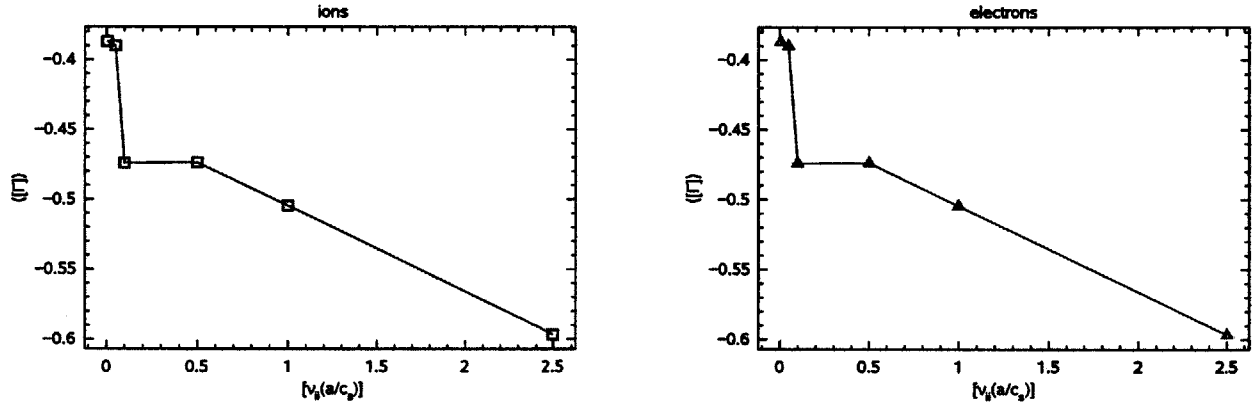


Figure 11: Variation of time-averaged particle flux with vnewk$_{(ii)}$.

Energy exchange (vnewk$_{(ii)}$ set)

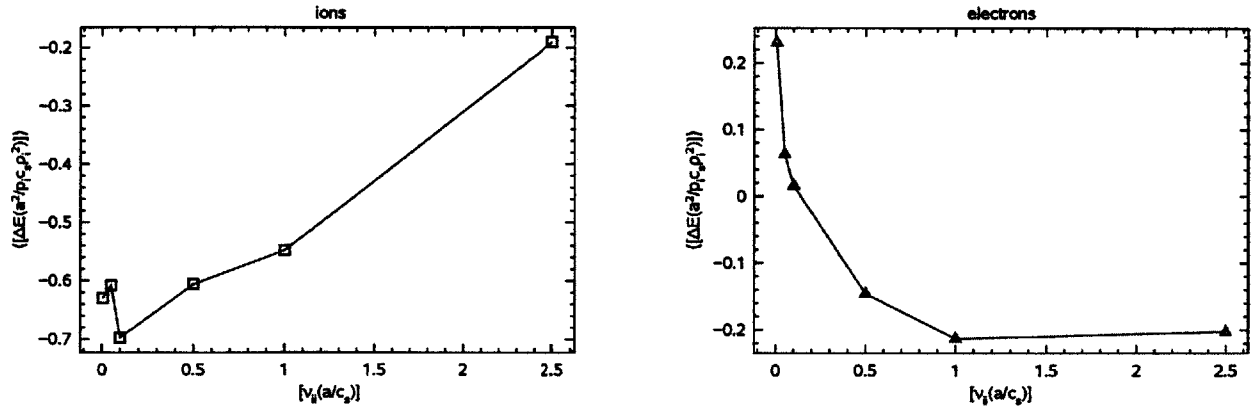


Figure 12: Variation of time-averaged energy exchange with vnewk$_{(ii)}$.

3.3 The collisional heating diagnostic

To assess the collisional heating diagnostic, an additional set of runs (with identical values set for $\texttt{vnewk}_{(ee)}$ to that of the previous section) were executed, but with the heating flag switched on. The expected behavior was that the diagnostic has no significant effect on any of the turbulent fluxes.

Context

Collisional heating is a measure of the irreversible heating of the equilibrium via collisional dissipation of the fluctuating part of the distribution function of a given species, and is thus related to the rate of entropy generation within the system [6]. The heating calculated this way is therefore dependent on the way that this dissapation is numerically modeled—in GS2, it is:

$$Q_{\text{col}} \equiv -\overline{\int d^3\mathbf{v} \frac{h_s T_s}{F_{0,s}} \langle C[h_s] \rangle_{\mathbf{R}}}, \tag{19}$$

where the integral is over velocity space and the overline represents a spatial average. The activation of the collisional heating diagonstic determines whether or not this term is included in the energy balance equation used by GS2, which otherwise includes terms describing energy transport, energy added to turbulence via background inhomogeneity, and temperature equilibration due to inter-species collisions.

Setup

This trial modeled electrons kinetically, used the Cyclone base case geometry, and had identical inputs to the aforementioned set of runs in which $\texttt{vnewk}_{(ee)}$ was varied. Each run recorded output quantities at 1000 time-steps so that their time histories could be compared against the corresponding runs in which the collisional heating diagnostic was switched off.

Results

An example heat flux time trace comparison is shown in Fig. 13. It can be seen that for lower values of electron collisionality parameter, the collisional heating diagnostic had a small, but visible difference on the resulting turbulent fluxes. As the collisionality parameter increased, this difference in value decreased, and for the highest value tested ($\texttt{vnewk}_{(ee)} = 2.5$), the time traces of the turbulent fluxes showed no difference in response to activating the heating diagnostic. Since activating the diagnostic implies that additional calculations are made at each step, the runs in which collisional heating was computed tended to reach shorter normalized times in the same number of timesteps as compared to those runs in which the diagnostic was left inactive.
The most notable result involves a significant difference between the particle and momentum fluxes as calculated with the diagnostic active and that with the diagnostic inactive: the time-averages differ in sign, with the activation of the diagnostic corresponding to a negative time-averaged particle flux and the run with the inactive diagnostic corresponding to a positive value; the momentum flux follows the same pattern. This only occurs for $\texttt{vnewk}_{(ee)} = 0.05$ and is immediately apparent from the particle flux time trace (and is not as easy to distinguish from the momentum flux time trace). At all other tested $\texttt{vnewk}_{(ee)}$ values, despite any resulting differences in time histories or time-averages, the collisional heating diagnostic did not have the effect of changing the sign of any time-averaged quantities.

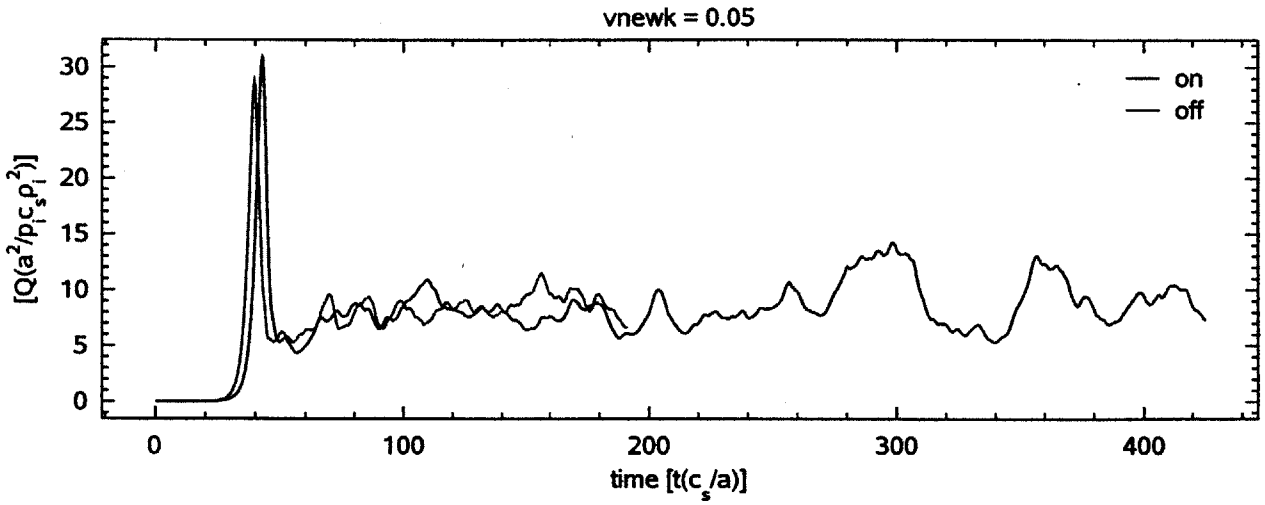


Figure 13: time trace comparison of turbulent ion heat flux between having the collisional heating diagnostic switched on and off. The difference between the red and blue traces tended to decrease as $\texttt{vnewk}_{(ee)}$ increased.

Analysis

From these time trace data, the time-averaged fluxes at each tested value of electron collisionality parameter can be computed. Averages for the present analysis are obtained by excluding the initial startup phase that can be recognized in each time trace plot of ion heat flux, only including the time during which GS2 is modeling turbulent phenomena. The normalized time ranges defined in this way are shown in Table 9.

Table 9: Normalized time ranges over which average values are computed. These ranges are analagous to the shaded region of the example time trace, corresponding to the post-startup turbulent behavior of interest.

$\texttt{vnewk}_{(ee)}$	Normalized time range
0.005	49.07 – 192.04
0.05	59.03 – 191.06
0.1	69.12 – 335.37
0.5	79.29 – 346.16
1.0	89.25 – 348.00
2.5	99.25 – 381.68

The collisional heating diagnostic can have a potentially non-negligible effect on GS2's calculated fluxes. This effect of a slightly higher or lower resulting quantity is more pronounced at lower values of electron collisionality parameter: each plot in appendix C shows significant deviation between red and blue traces for tested values of this parameter up to 0.1 (after which the difference is more slight), and at $\texttt{vnewk}_{(ee)} = 2.5$, the largest probed value, the time histories of the runs with the diagnostic off and the diagnostic on are not visibly distinguishable. This relative difference attributable to the diagnostic (as well as its dependence on electron collisionality parameter) can be seen in Fig. 14.
Aside from the exceptional cases of the momentum and particle fluxes differing in sign at $\texttt{vnewk}_{(ee)} = 0.05$, each time-averaged quantity differed as a result of the diagnostic by less than 5% for all

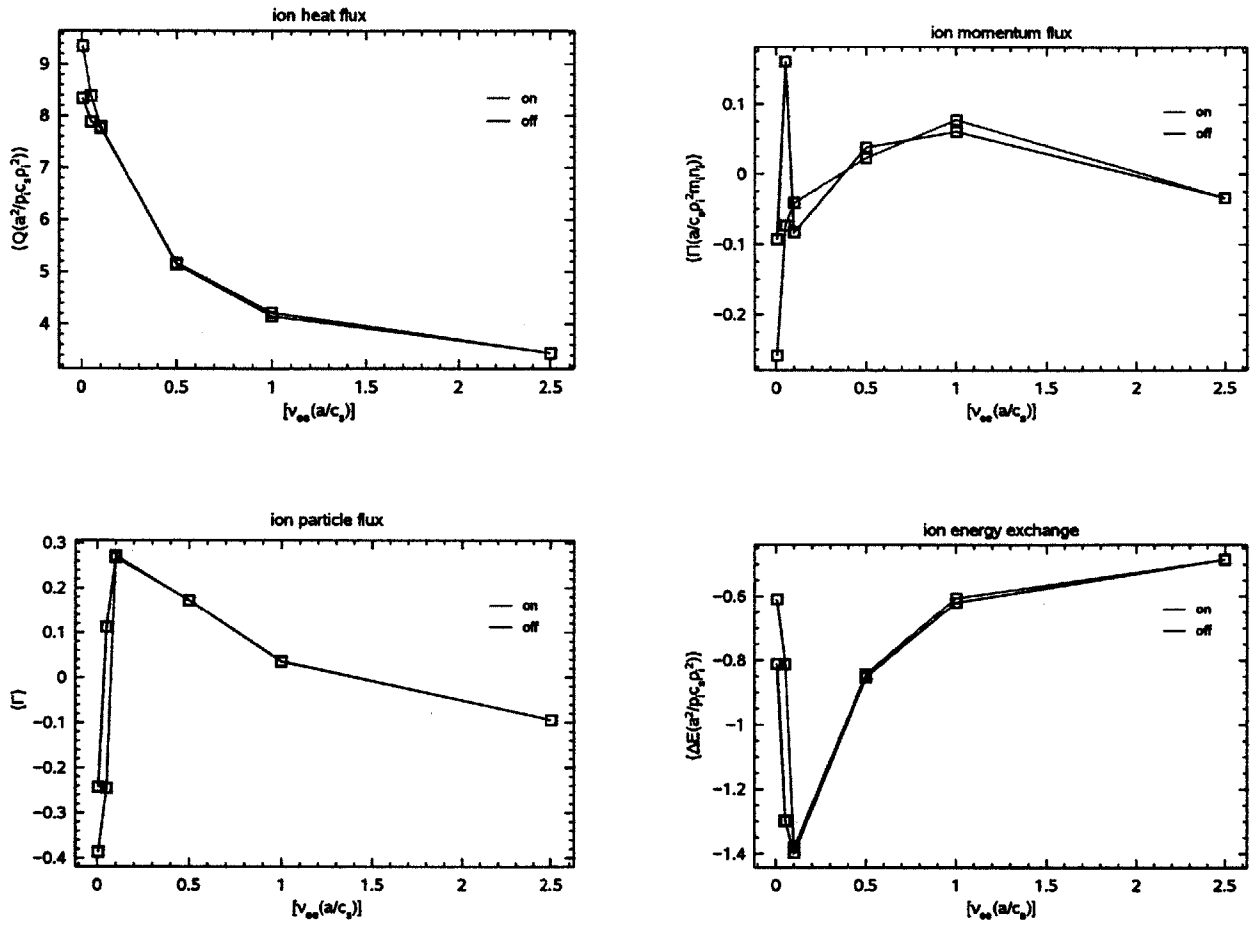


Figure 14: Comparison of time-averaged ion fluxes as calculated with the collisional heating diagnostic on and off.

tested values of electron collisionality parameter.

4 Discussion & conclusions

The trials presented above point out several key features of the connection between turbulent transport and intra-species collisionality. First, it may be useful to characterize the effects of $\mathbf{E}\times\mathbf{B}$ flow shear to contextualize those of collisionality. As found in the study used for reference and qualitatively confirmed in trial A, the effects of the flow shearing rate parameter are to suppress processes that drive turbulent heat, momentum, and particle flux as well as the energy exchanged between species. It acts to produce modes within the plasma that either grow or decay as they travel through regions of (un)favorable magnetic curvature, ultimately leading to an optimal value of shearing rate γ_E at a given temperature gradient length scale L. The implications are as follows: if implemented correctly (i.e. if high $\mathbf{E}\times\mathbf{B}$ shearing rates are achieved before the temperature gradient is allowed to increase such that it drives a prohibitive amount of turbulence), these optimal conditions correspond to minimal heat fluxes, greater stability, and enhanced confinement, critical factors in improving the operation of magnetic fusion devices. Characterizing the effects of collisionality is less straightforward—it does not appear to have as consistent an effect on turbulent fluxes. While each of the flux quantities presently considered generally grew closer to zero as both collisionality parameters increased, the form of the responses of heat, momentum, and particle fluxes differed greatly.

The heat flux had a much more pronounced response to electron-electron collisionality, decreasing in value by nearly 1/2 over the range of tested collisionalityes (the corresponding decrease in response to ion-ion collisionality was only about 1/5). This decrease appears to be monotonic, with most of the change occuring as g_exb increases from 0 to 1. Its response to electron-electron and ion-ion collisionalites are qualitatively similar to one another and to its response to flow shearing, though the mechanisms behind each observed response most likely differ vastly.

The response of the momentum flux was was also similar between varying ion-ion and electron-electron collisionality—the general trend was a decrease in magnitude, but there seem to be a number of local minima and maxima within this range. There was an important difference between the responses to varying different collisionalities: momentum flux was positive-valued throughout the variation of electron-electron collisionality parameter, while it alternated between positive and negative values (as it decreased in magnitude overall) when ion-ion collisionality was varied. It appears that a net inwards or outwards transport of toroidal angular momentum (with respect to a given flux surface) is possible at a number of different collisionalities. A possible explanation is that turbulent momentum transport is determined by a multitude of processes that are driven by particle collisions to varying degrees; there may exist certain ranges of collisionality parameter to which certain mechanisms that contribute toward or suppress the transport of momentum are more sensitive.

The effect on particle flux differed significantly between varying electron-electron and ion-ion collisionality parameters. For the electron-electron case, there was a single local (positive-valued) maximum at a relatively small value of collisionality, a steep increase from negative values to this maximum at low collisionality, and a more gradual decrease back down to negative values beyond the maxmimum. These results seem to identify a certain undesirable range of electron-electron collisionality associated with the outward transport of particles and conversely a net inward transport of particles for values of collisionality outside of this range. This could mean that for better fusion performance, electron-electron collisionality must either be zero or very high in magnitude. In response to ion-ion collisionality, on the other hand, the particle flux was never observed to reach positive values. In fact, it decreased seemingly discontinuously as the parameter increased from zero before remaining relatively constant over a short range and then gradually decreasing as the parameter increased further. This steady increase in magnitude of the

(negative) particle flux as both electron-electron and ion-ion collisionality increases associates high rates of both ion-ion and electron-electron collisions with better fusion performance. Turbulent energy exchange was also affected differently by the variation of electron-electron and ion-ion collisionality. The discontinous increase and then somewhat more gradual decrease in magnitude as electron-electron collisionality is raised from zero makes intuitive sense: as electron-electron collisions become more dominant, the net change in energy attributable to electron-ion collisions becomes less imporant. While there is a simlilar—albeit monotonic—decrease in magnitude in the exchanged energy as ion-ion collisionality is increased (for presumably similar reasons), there is an interesting behavior that occurs at low values of this parameter. There does not appear to be a discontinuous change as it increases from zero, and the results suggest that electrons can actually *lose* energy to ions at high enough values of ion-ion collisionality.

It was found through use of the diagnostic that collisional heating only has observable effects on turbulent fluxes at low electron-electron collisionalities, and only slightly at that. From this it may be concluded that the collisional heating term in the energy balance equation implemented in `GS2` can be appropriately neglected in most investigations performed with the code. However, the anomalous difference in sign between predicted momentum and particle fluxes at a particular value of electron-electron collisionality parameter emphasizes the difficulty of reliably capturing the relevant physics of charged particle collisions using a numerically implementable collision operator. As it is not possible for a full-fidelity representation, the challenge becomes a matter of using and developing ever-improved approximations—this reflects the basic limitations of computational research and is a well-understood factor.

The markedly different responses of turbulent heat, momentum, and particle transport to changes in both electron-electron and ion-ion collisionalities serve to demonstrate the sheer complexity and sensitivity to operating parameters of magnetically confined plasmas. Certain behvaiors of each quantity can be identified for values of collisionality parameter close to zero, values close to unity, and those that are much higher, but the details of those behaviors as well as their underlying mechanisms are not common to each quantity. In this sense, the findings presented here agree on some level with those obtained by simulating conditions within Alcator C-Mod with `GYRO` [4]; these results suggest that all of the examined turbulent quantities are associated with a number of different drives, only some of which are shared universally. Further characterizing the effects of this intricate set of dependencies (e.g. by asking the question: which range of values of a certain parameter, if any, is associated with turbulent heat, particle, and momentum transport properties that are simultaneously desirable?) is a crucial step in ultimately determining an optimal set of physical parameters for the production of magnetic fusion power.

Further work

Within the scope of the present investigation, there are several immediate avenues of exploration. One possibility would be to conduct similar trials at various values of ion temperature and density gradients (i.e. additionally varying collisionalities at each of a specified set of `tprim` and `uprim` values). This would allow a more general idea of collisionality's effect on turbulence to be gained. Also increasing the resolution of the time-average analysis by probing additional values of `vnewk` would help clarify the trends found in the time-average plots; this would be especially beneficial for analyzing the momentum flux results. The diagnostic could also be tested with ion-ion collisions to see if that associated heating term is significant in any region of paramter values. Finally, and perhaps most importantly, a numerical investigation (possibly using an alternative gyrokinetic code) in which the ion-electron collisionality is directly controlled could potentially

reveal useful physics and help contextualize the results from intra-species collisionality variation.

References

[1] M. Barnes et al. "Turbulent Transport in Tokamak Plasmas with Rotational Shear." Physical Review Letters (2011). http://dx.doi.org/10.1103/PhysRevLett.106.175004

[2] J. E. Kinsey et al. "The effect of safety factor and magnetic shear on turbulent transport in nonlinear gyrokinetic simulations." Physics of Plasmas (2006). http://dx.doi.org/10.1063/1.2169804

[3] J. Candy. "Turbulent energy exchange: Calculation and relevance for profile prediction." Physics of Plasmas (2013). http://dx.doi.org/10.1063/1.4817820

[4] A. E. White et al. "Multi-channel Transport Experiments at Alcator C- Mod and Comparison with Gyrokinetic Simulations." Physics of Plasmas (2013). http://dx.doi.org/10.1063/1.4803089

[5] I. G. Abel et al. "Linearized model Fokker-Planck collision operators for gyrokinetic simulations. I. Theory." Physics of Plasmas (2008). http://dx.doi.org/10.1063/1.3046067

[6] M. Barnes et al. "Linearized model Fokker-Planck collision operators for gyrokinetic simulations. II. Numerical implementation and tests". *Phys. of Plasmas* (2008). http://dx.doi.org/10.1063/1.3155085

[7] G. W. Hammett. Gyrokinetic Theory and Simulation of Experiments. APS Div. Plasma Physics Orlando, 11/12/2007. Accessed 4/10/14. http://fire.pppl.gov/aps07_hammett_gyro.pdf

[8] W. Dorland. *GS2 Homepage.* Accessed 2/28/14. http://gs2.sourceforge.net

[9] G. G. Plunk. "The theory of gyrokinetic turbulence: A multiple-scales approach". (2009). http://arxiv.org/pdf/0903.1091v2.pdf

[10] M. Kotschenreuther, G. Rewoldt, W. M. Tang. "Comparison of initial value and envalue codes for kinetic toroidal plasma instabilities". *Computer Physics Communications* 88 (1995) 128-140. http://gs2.sourceforge.net/docs/kot95/kotschenreuther95.pdf

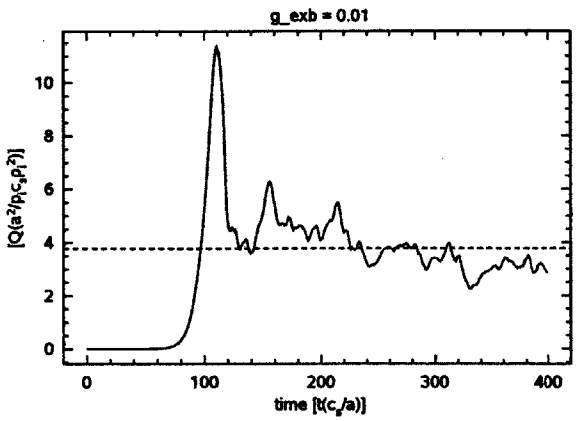
[11] W. Dorland. ITG: Cyclone base case. *Plasma Microturbulence Project Benchmarks.* Accessed 4/10/14. http://gs2.sourceforge.net/PMP/itg.html

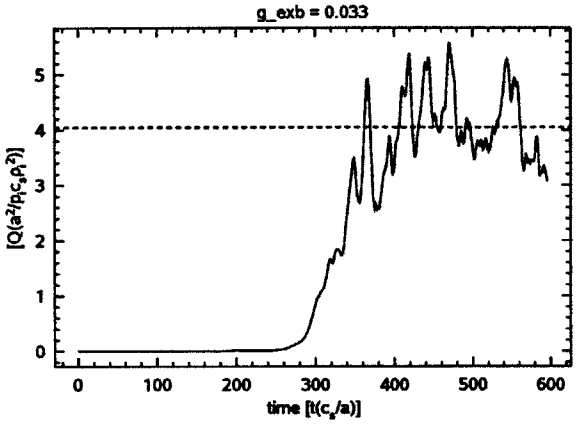
[12] J. Bezanson et al. "Julia: A Fast Dynamic Language for Technical Computing". (2012). http://arxiv.org/pdf/1209.5145v1.pdf

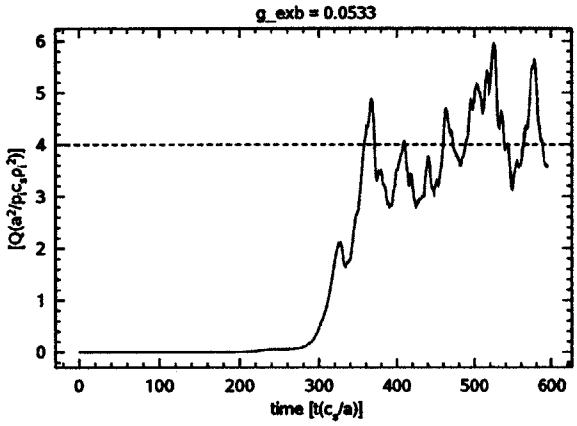
[13] M. Barnes. *GS2 Gyrokinetics Wiki.* Accessed 3/3/14. http://sourceforge.net/apps/mediawiki/gyrokinetics/index.php?title=Main_Page

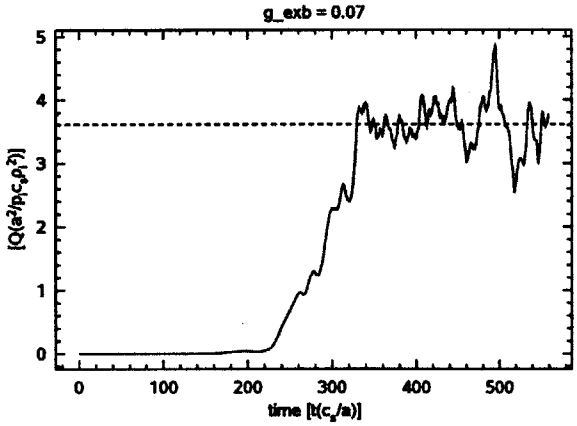
A Trial A time traces

Ion heat flux









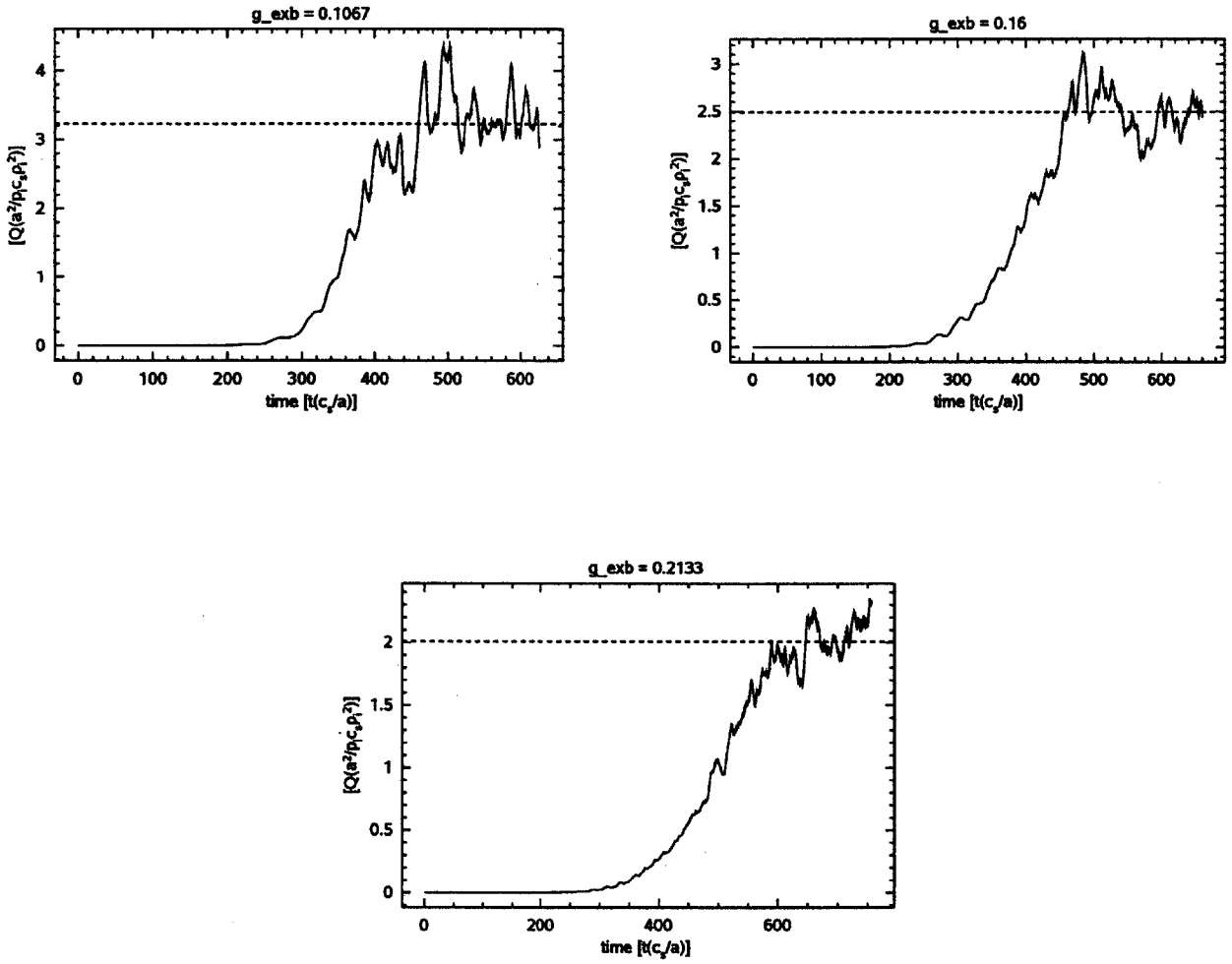
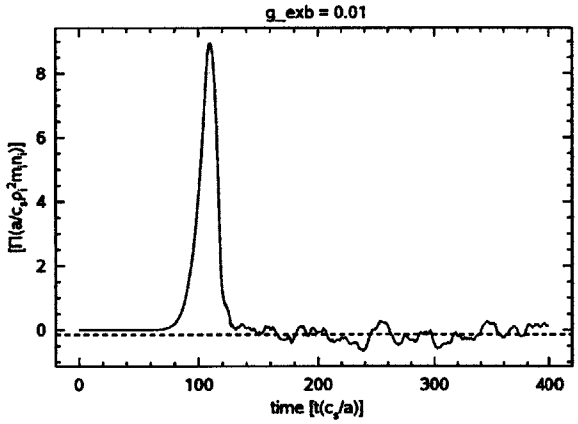
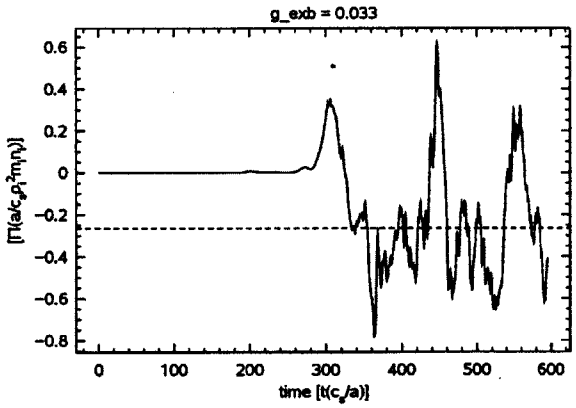
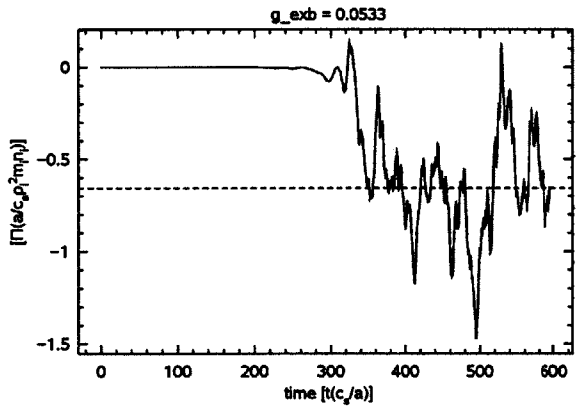


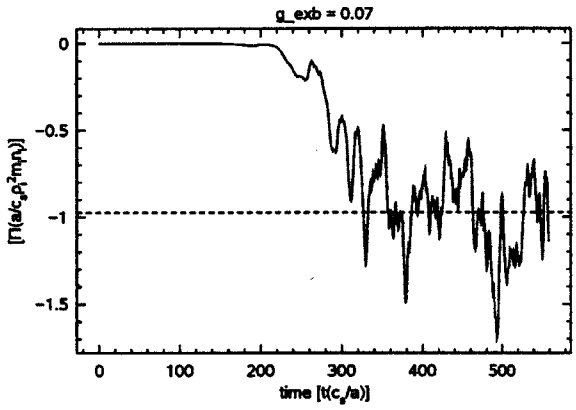
Figure 15: Calculated turbulent heat fluxes at each probed value of g_exb.

Ion momentum flux









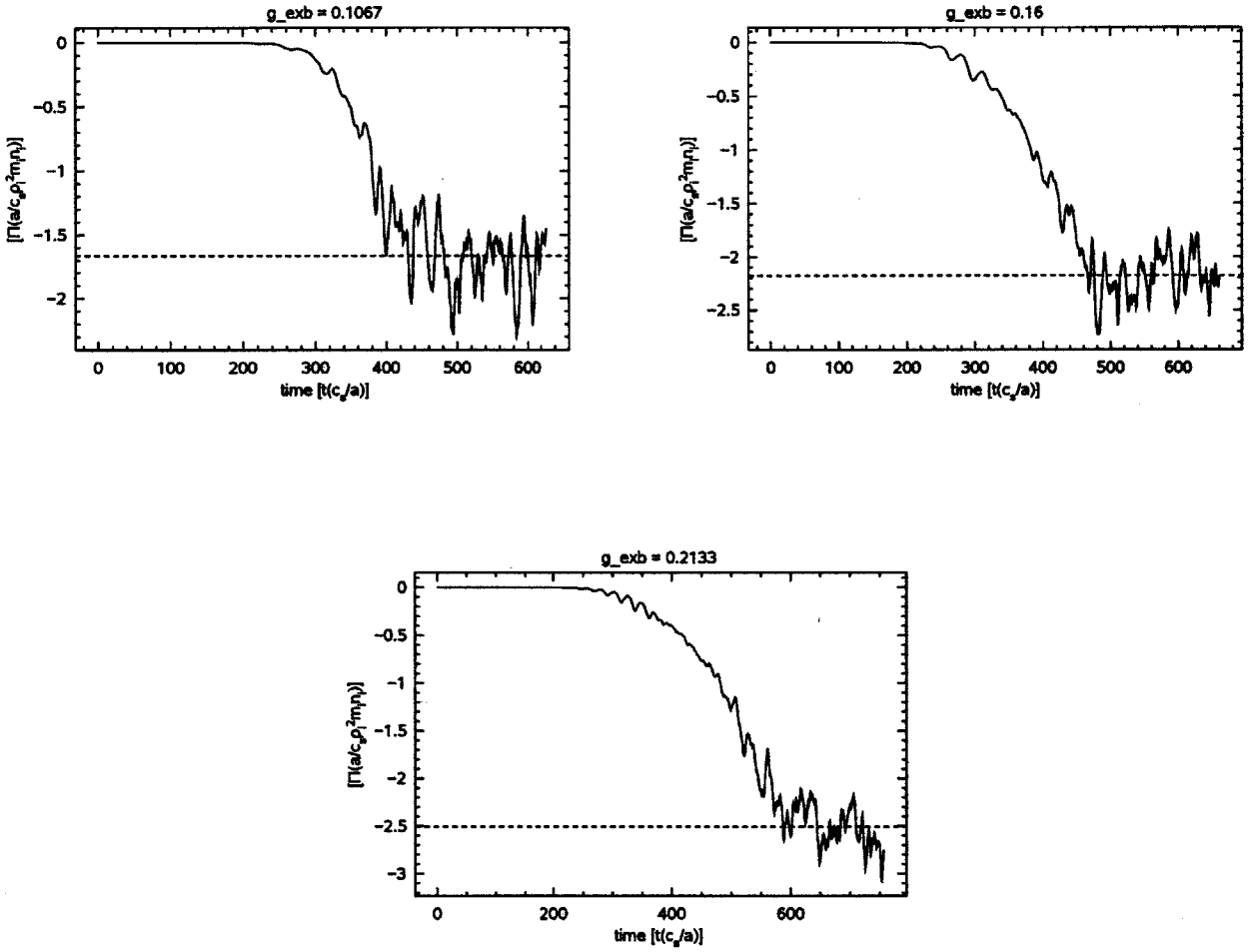
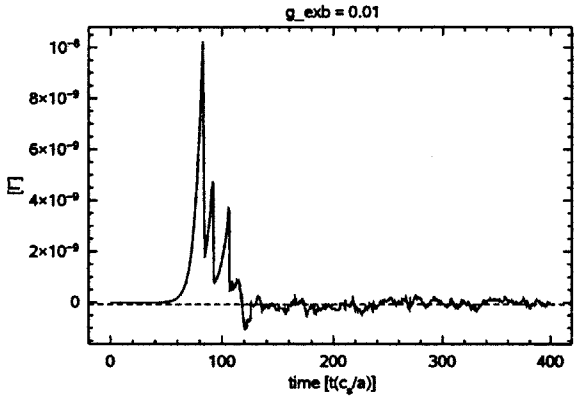
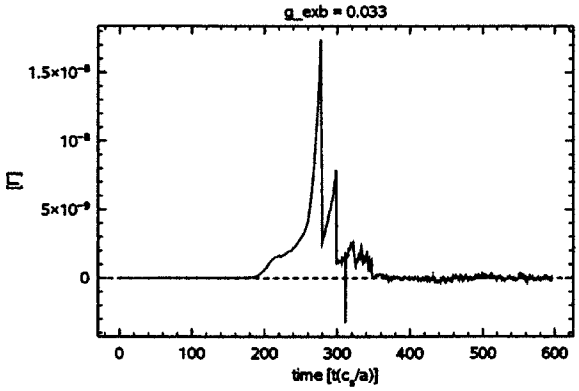
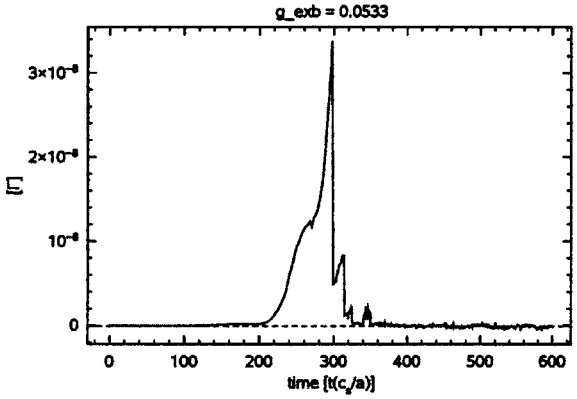


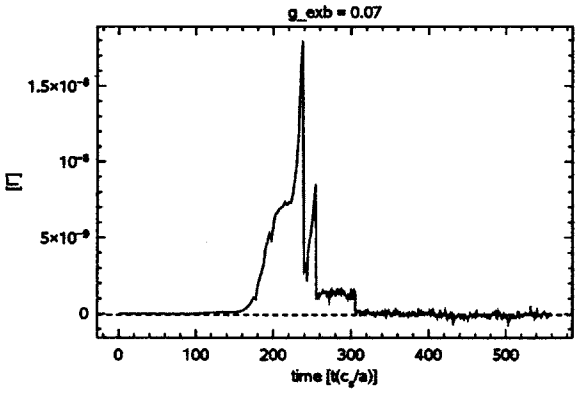
Figure 16: Calculated turbulent momentum fluxes at each probed value of g_exb.

Ion particle flux









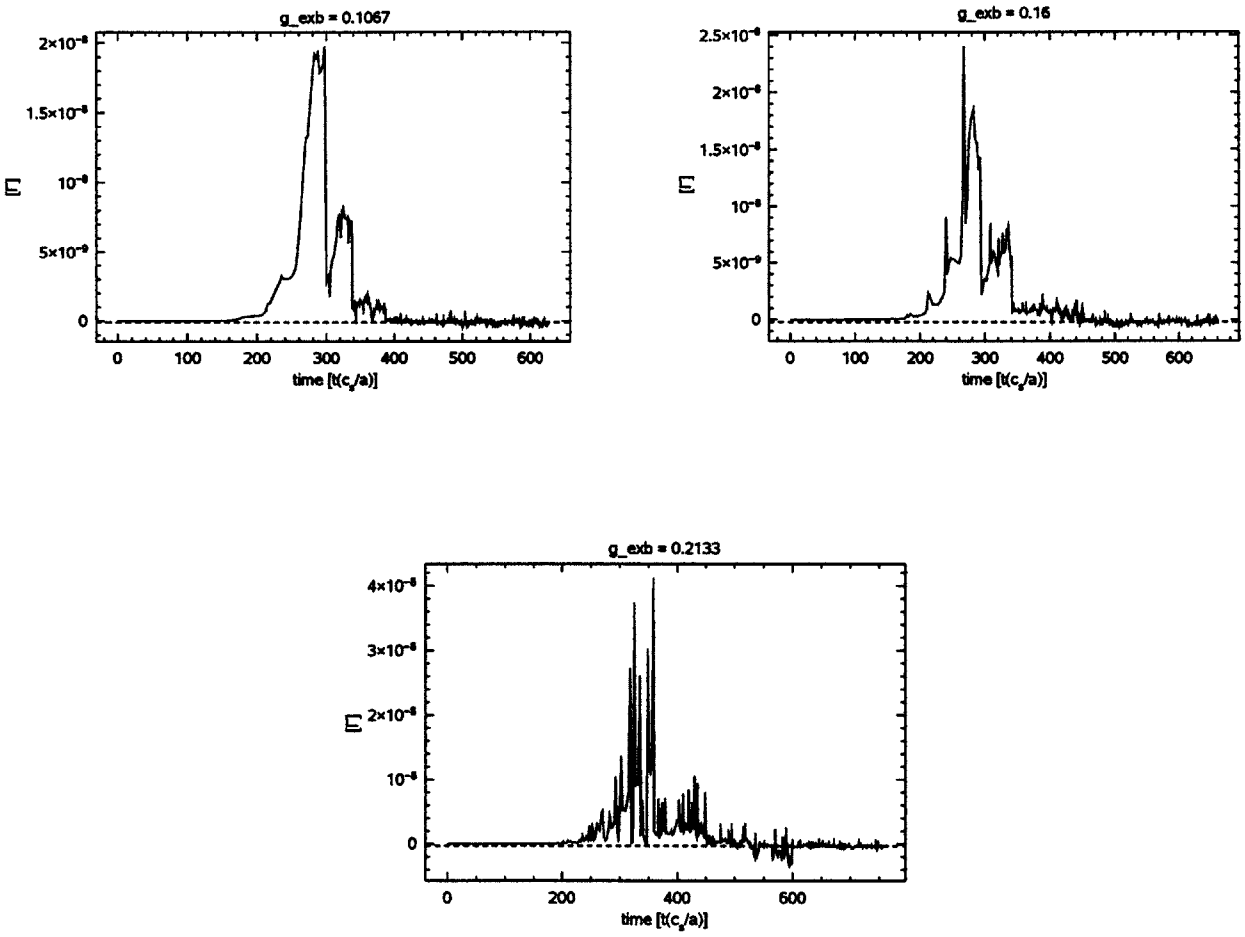
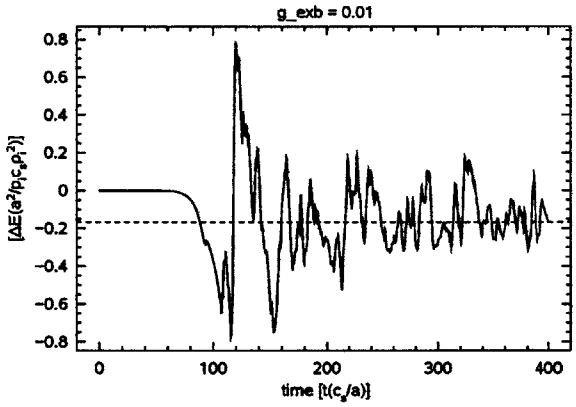
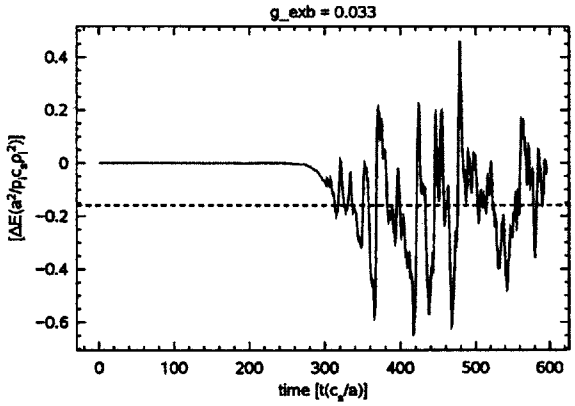
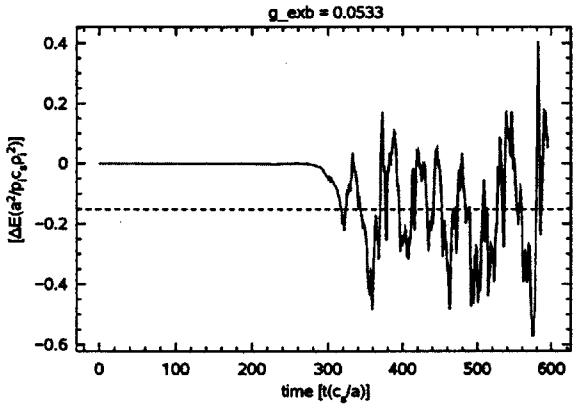


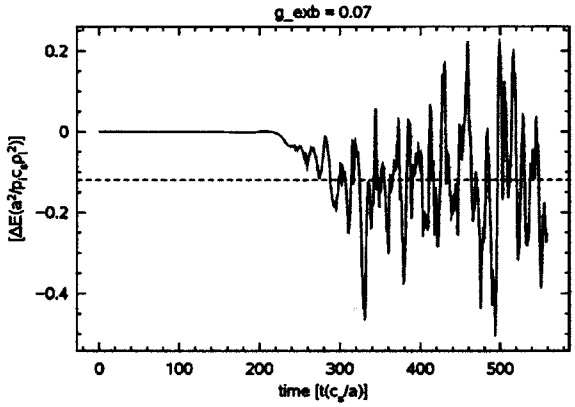
Figure 17: Calculated turbulent particle fluxes at each probed value of g_exb.

Ion energy exchange









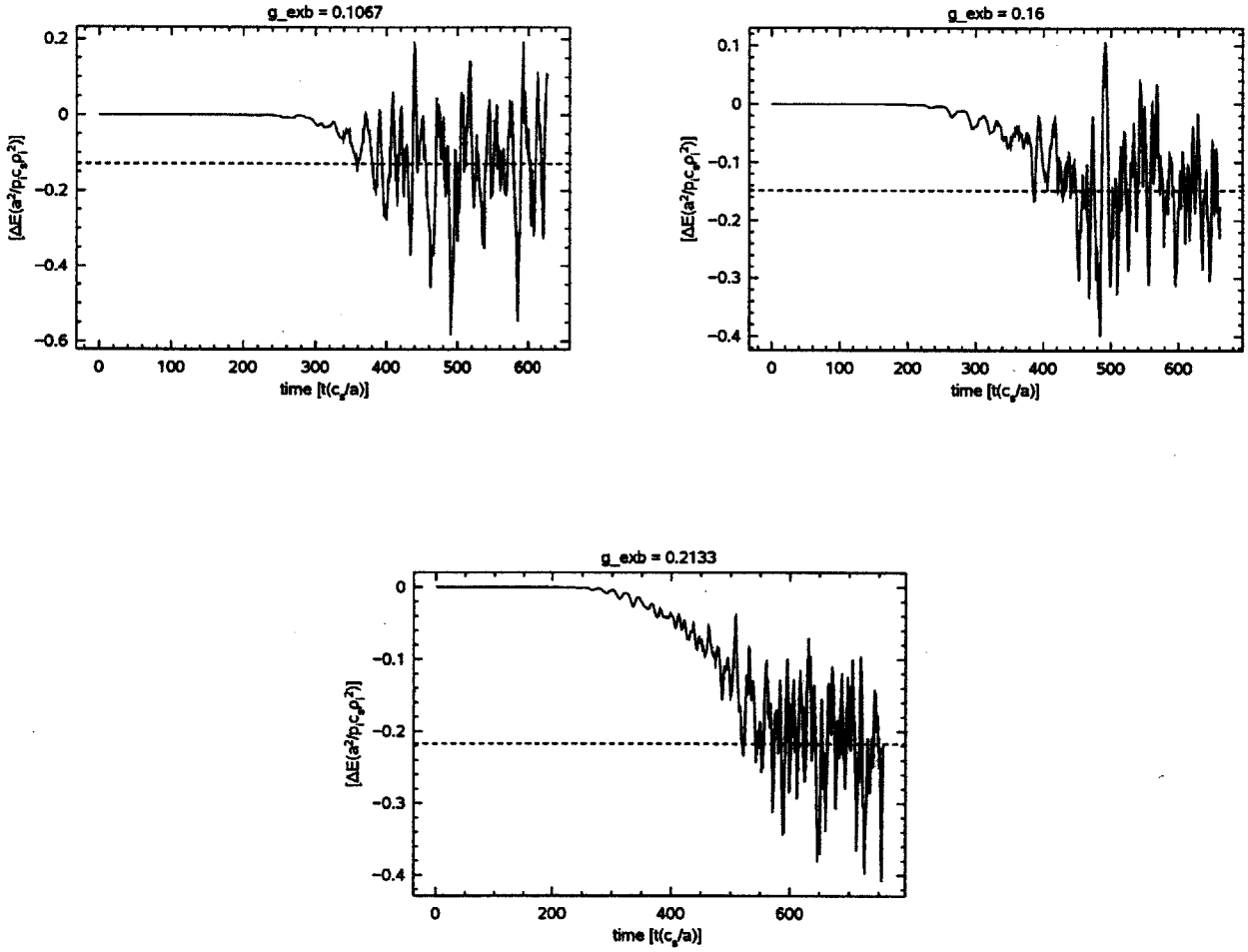


Figure 18: Calculated turbulent energy exchange at each probed value of g_exb.

B Trial B time traces

B.1 vnewk$_{(ee)}$ runs

Ion heat flux

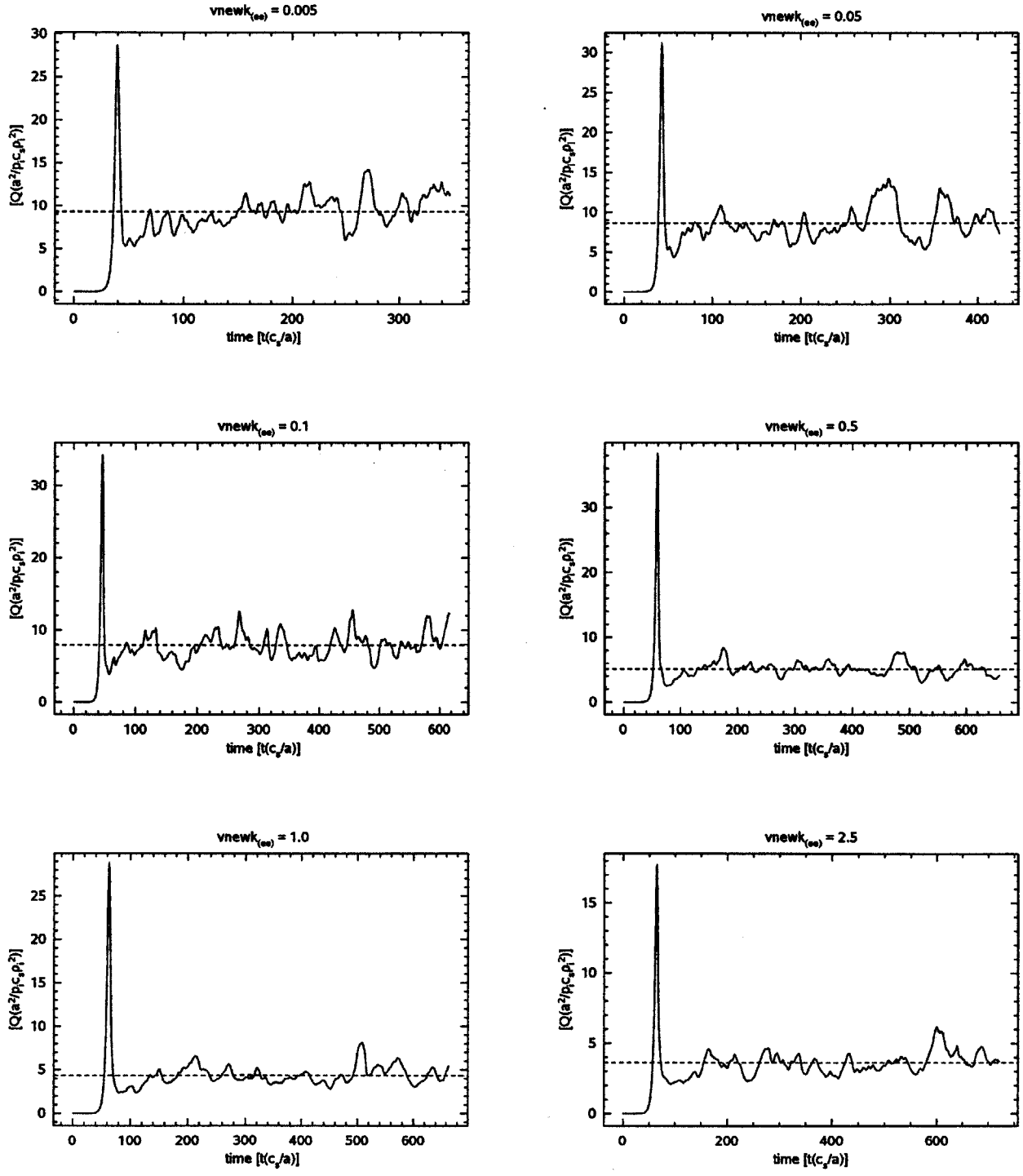


Figure 19: Ion heat flux at each probed value of vnewk.

Electron heat flux

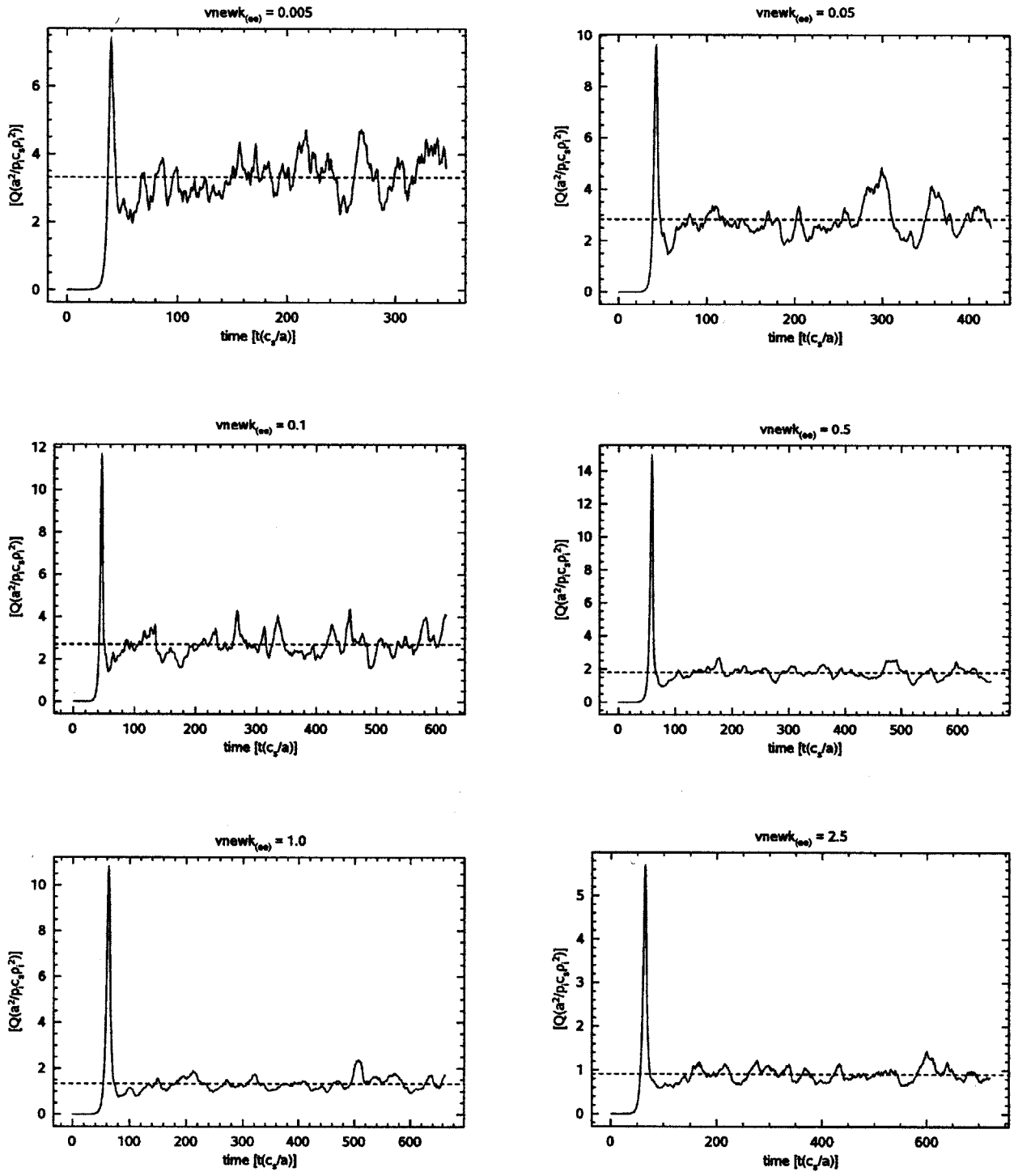


Figure 20: Electron heat flux at each probed value of vnewk.

Ion momentum flux

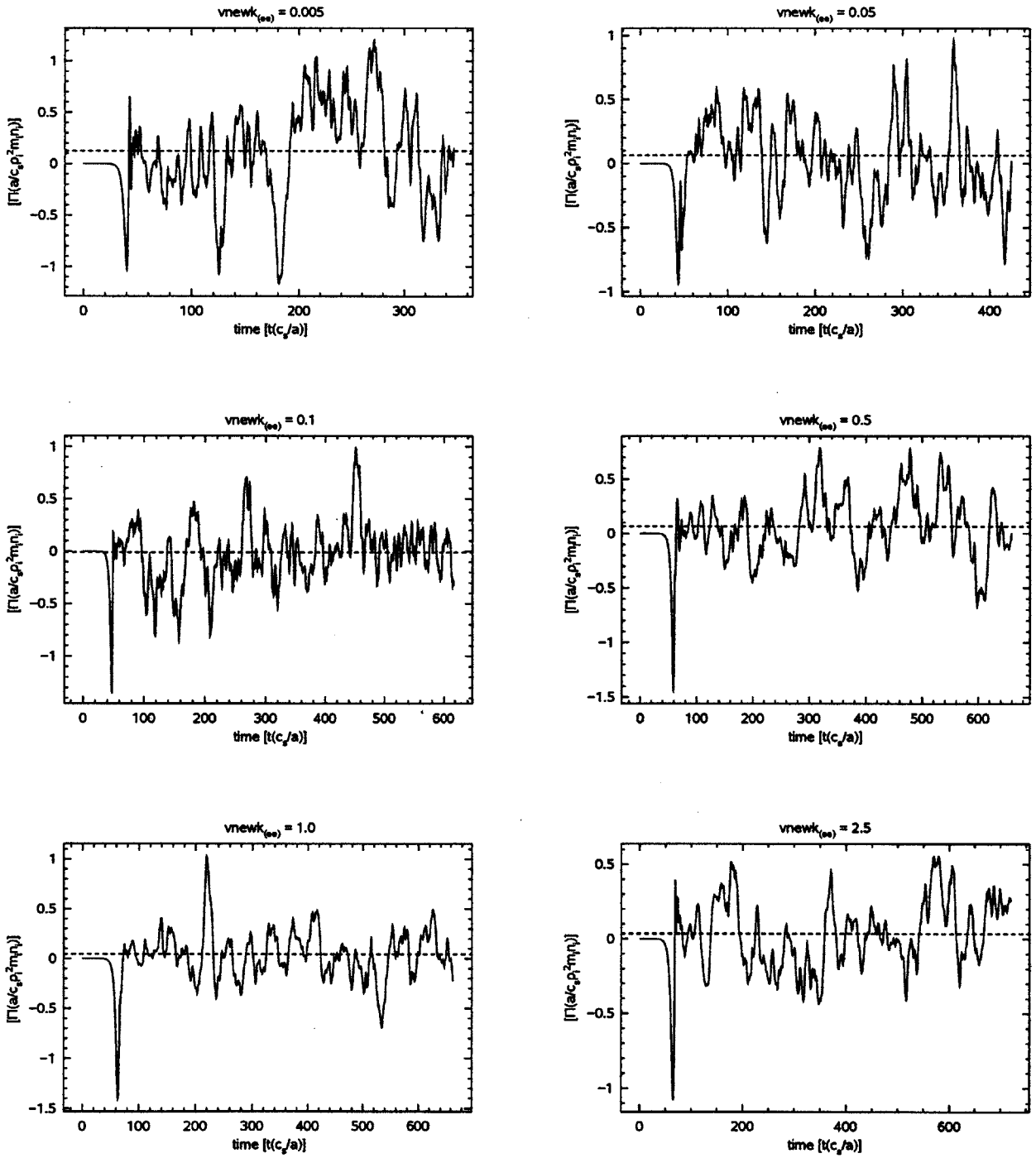


Figure 21: Ion momentum flux at each probed value of vnewk.

Electron momentum flux

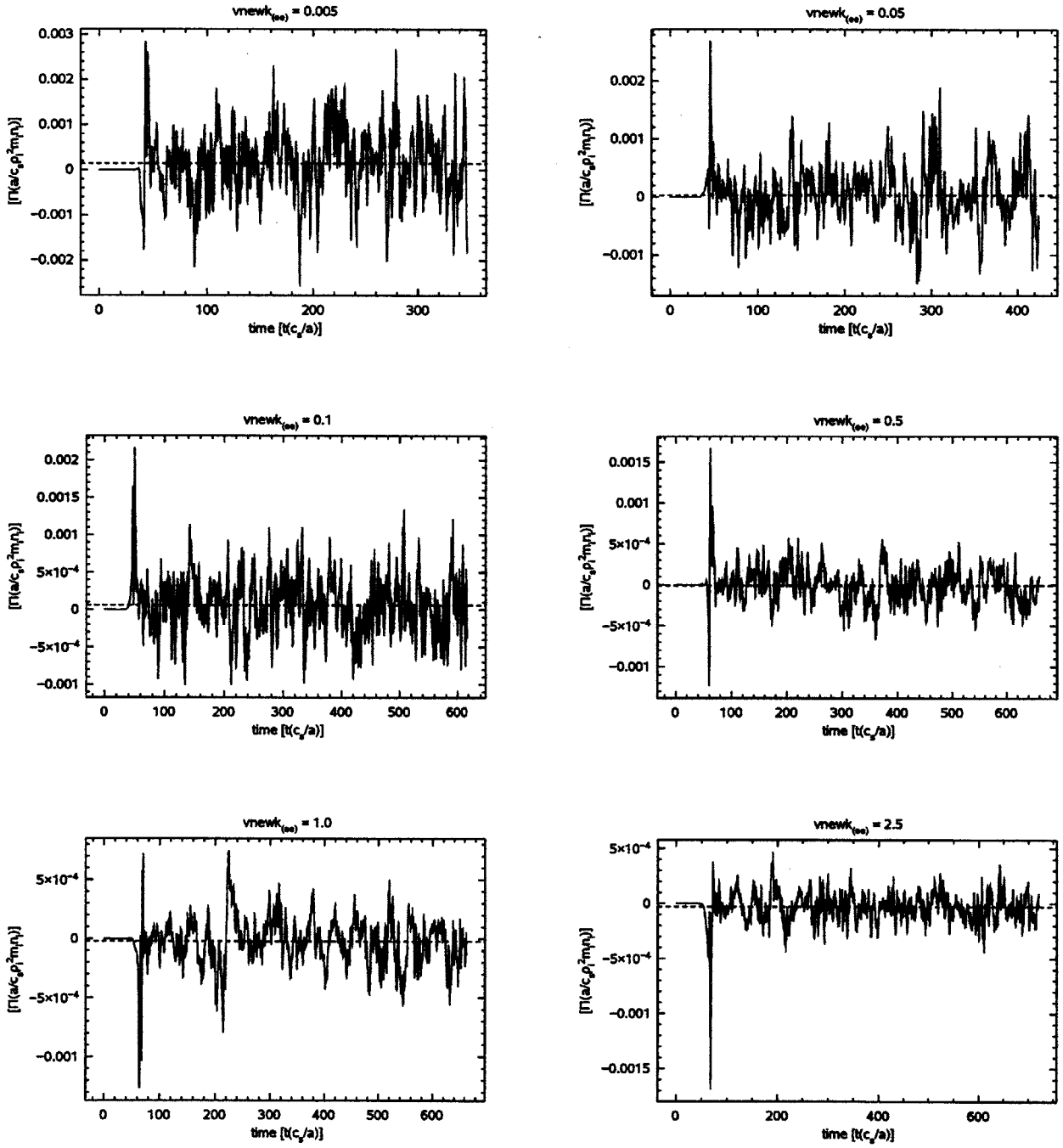


Figure 22: Electron momentum flux at each probed value of vnewk.

Ion particle flux

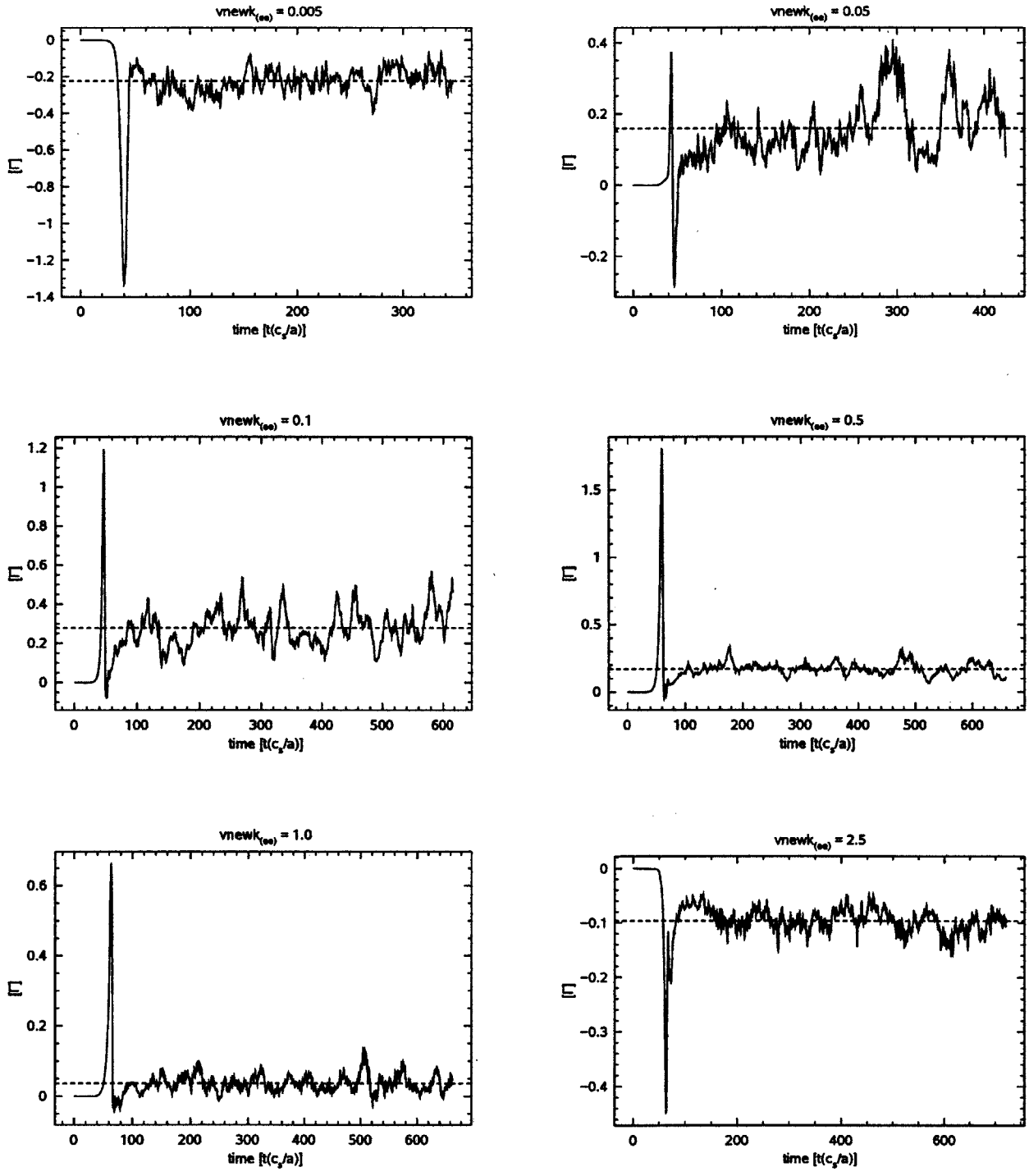


Figure 23: Ion particle flux at each probed value of vnewk.

Electron particle flux

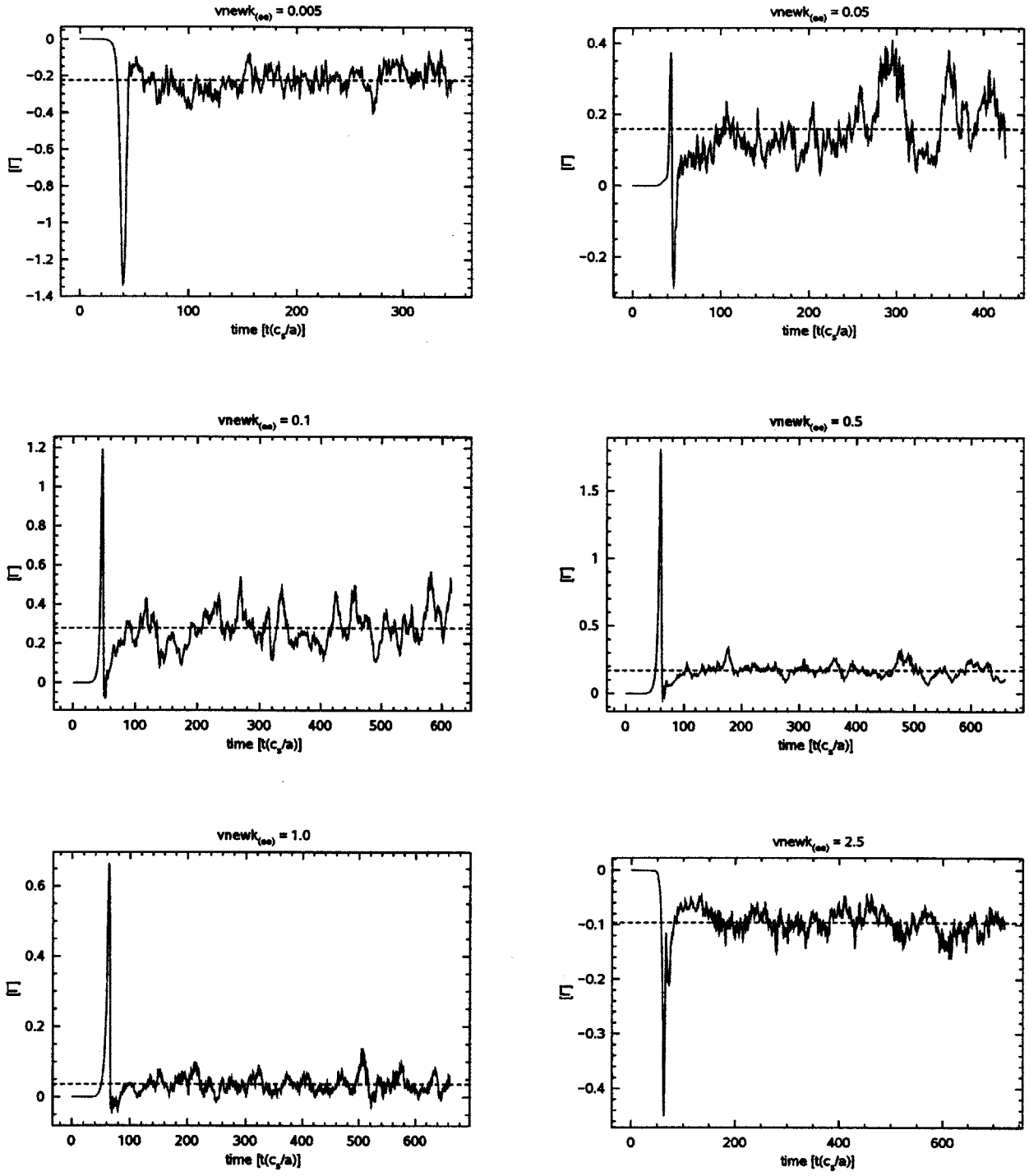


Figure 24: Electron particle flux at each probed value of vnewk.

Ion energy exchange

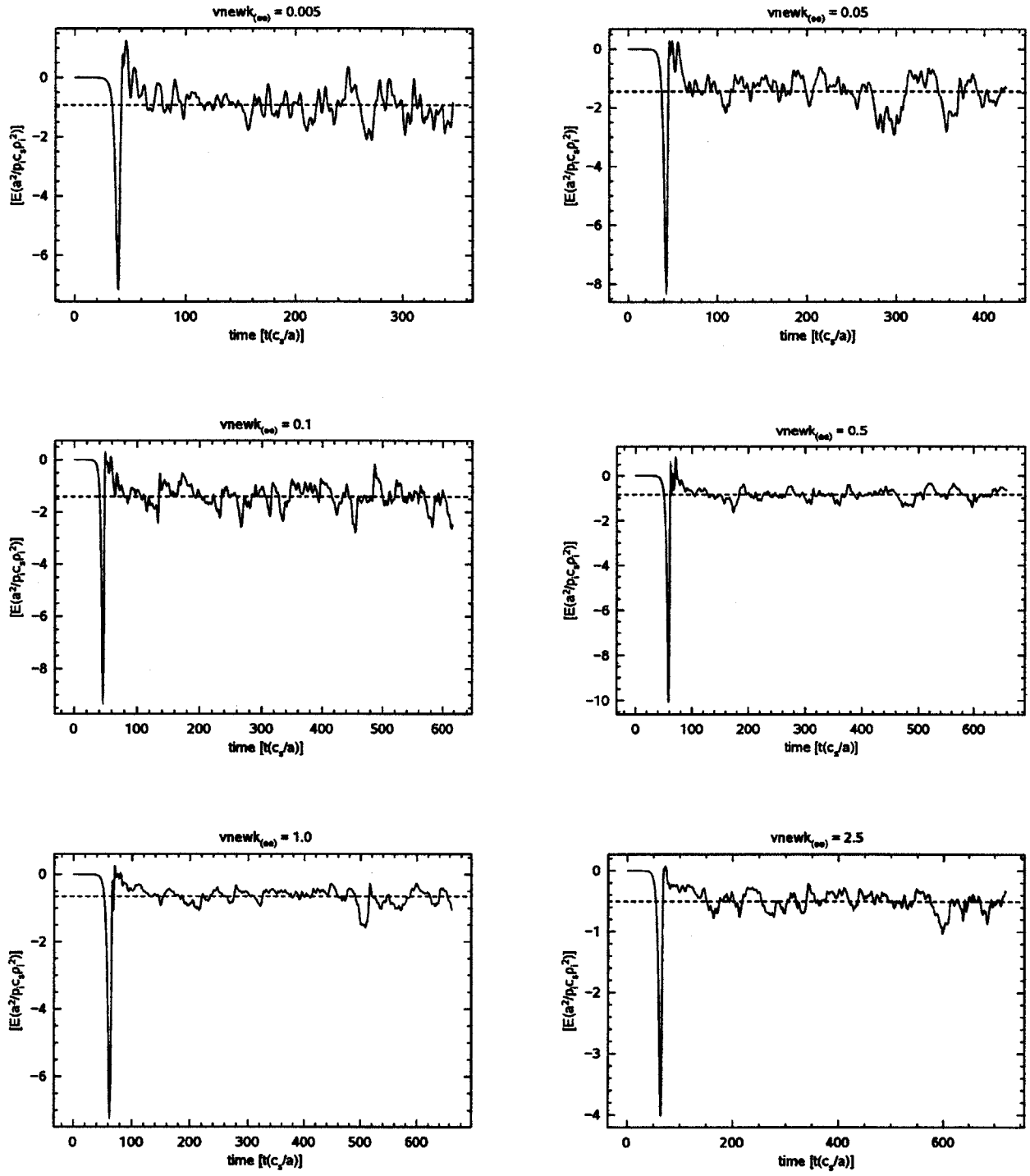


Figure 25: Ion energy exchange at each probed value of vnewk.

Electron energy exchange

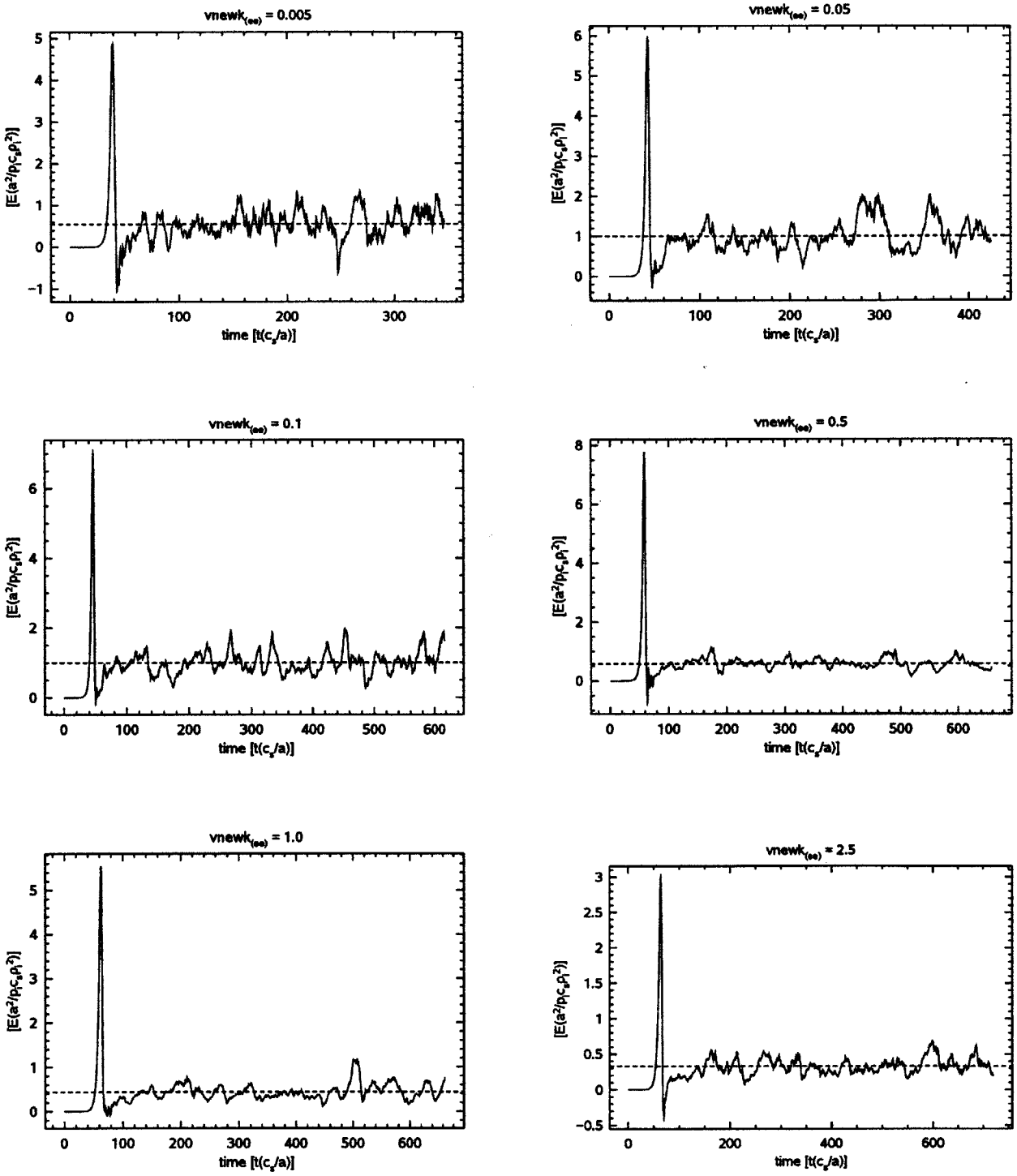


Figure 26: Electron energy exchange at each probed value of vnewk.

B.2 vnewk$_{(ii)}$ runs

Ion heat flux

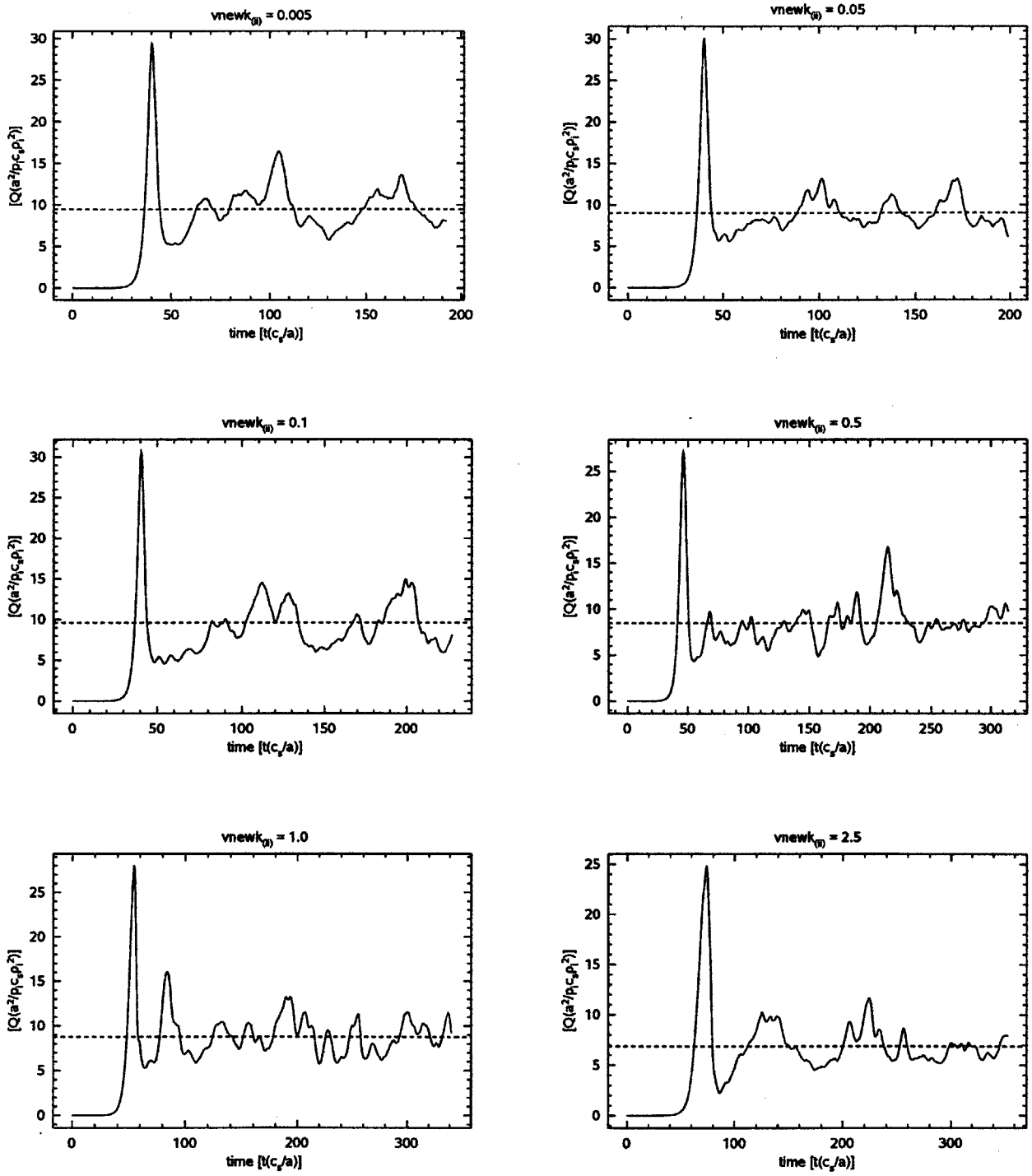


Figure 27: Ion heat flux at each probed value of vnewk.

Electron heat flux

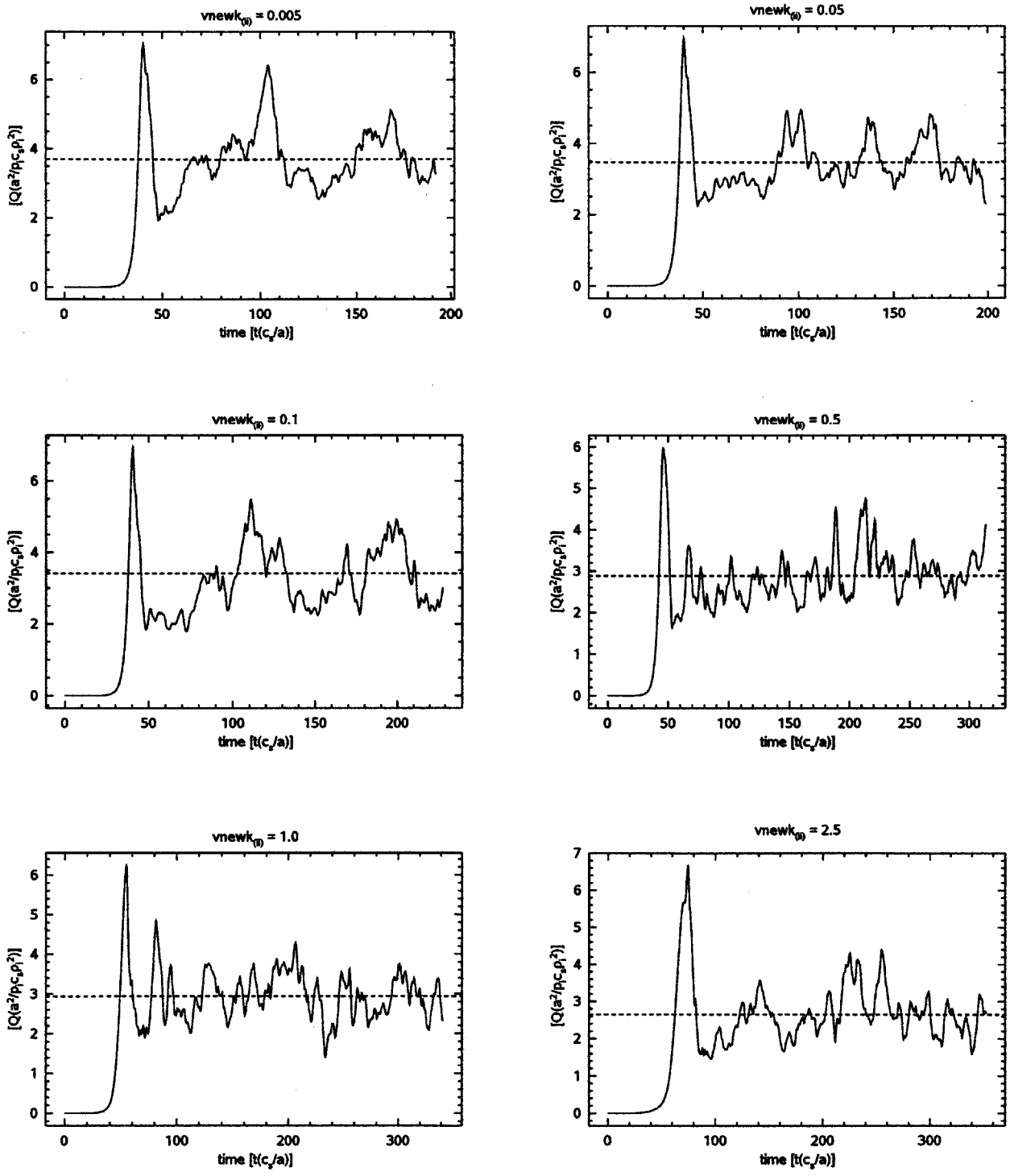


Figure 28: Electron heat flux at each probed value of vnewk.

Ion momentum flux

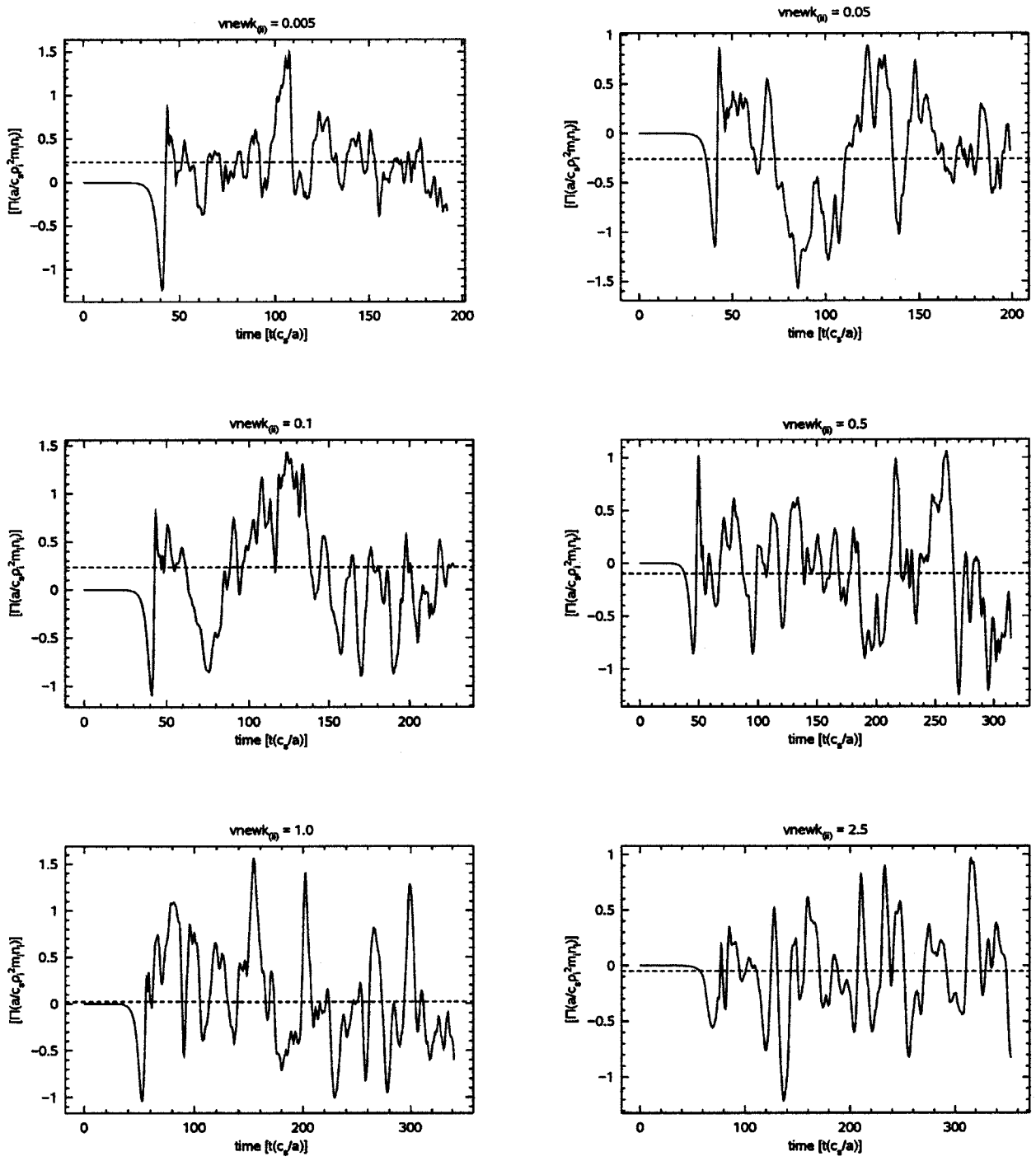


Figure 29: Ion momentum flux at each probed value of vnewk.

Electron momentum flux

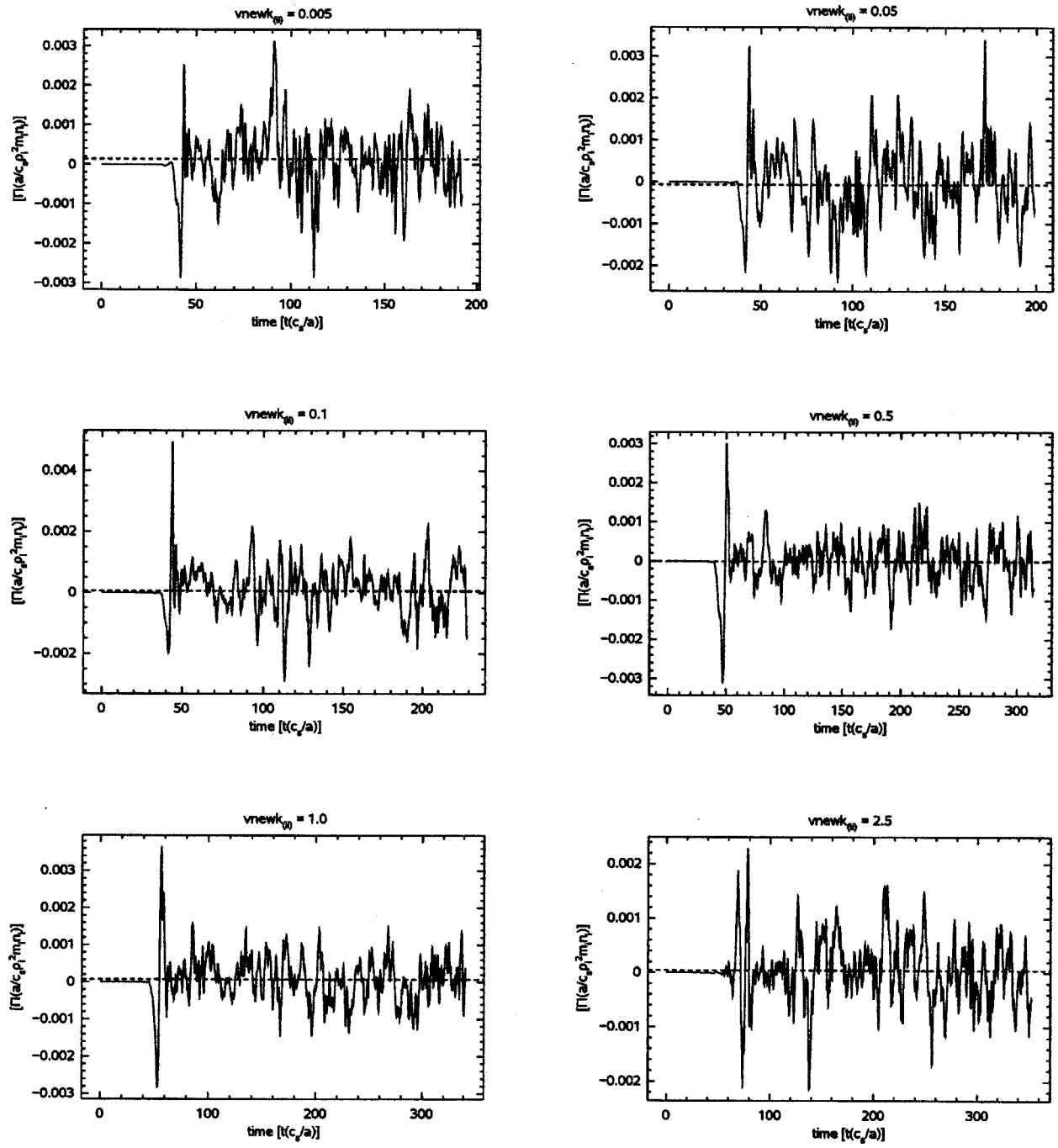


Figure 30: Electron momentum flux at each probed value of vnewk.

Ion particle flux

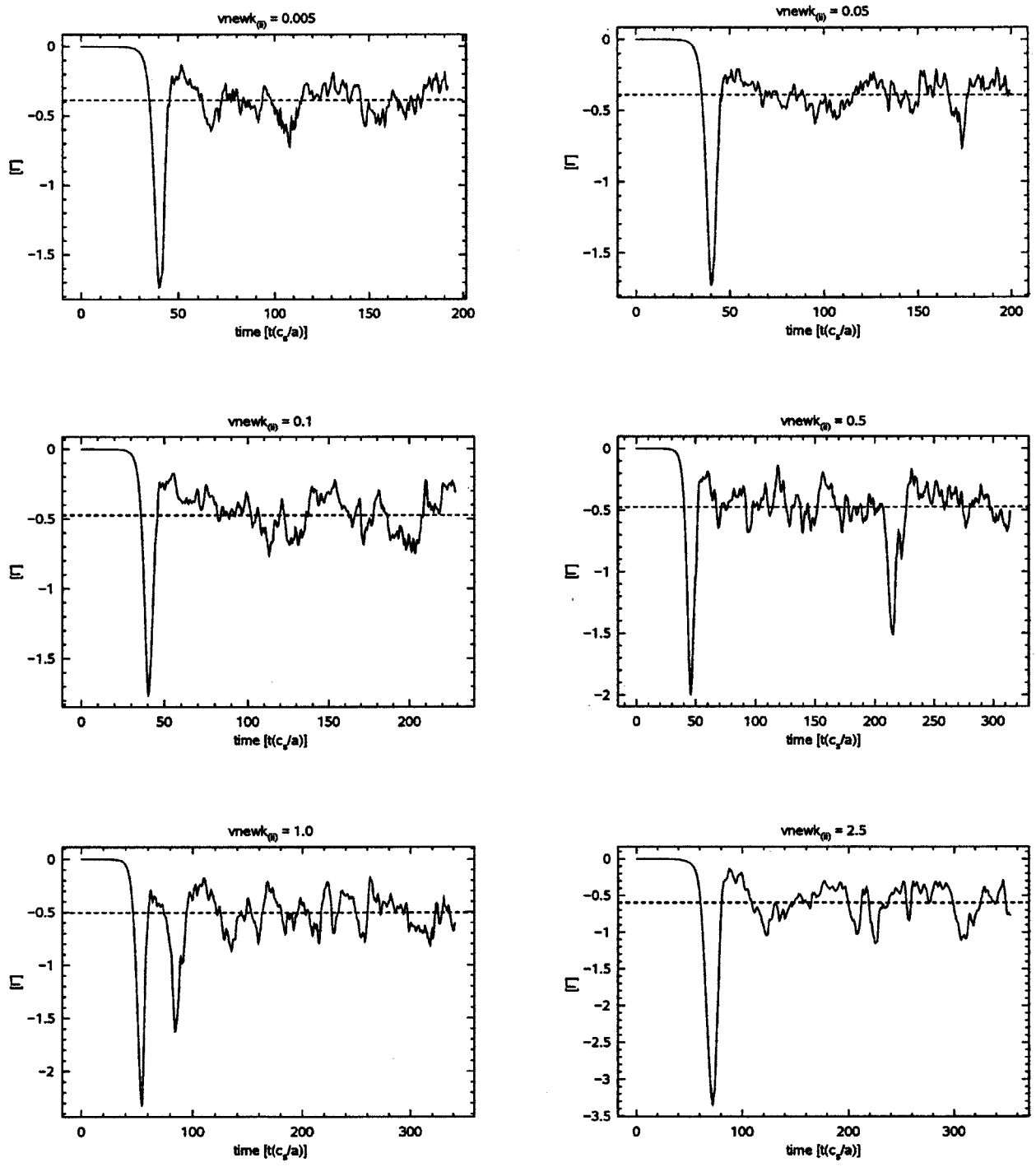


Figure 31: Ion particle flux at each probed value of vnewk.

Electron particle flux

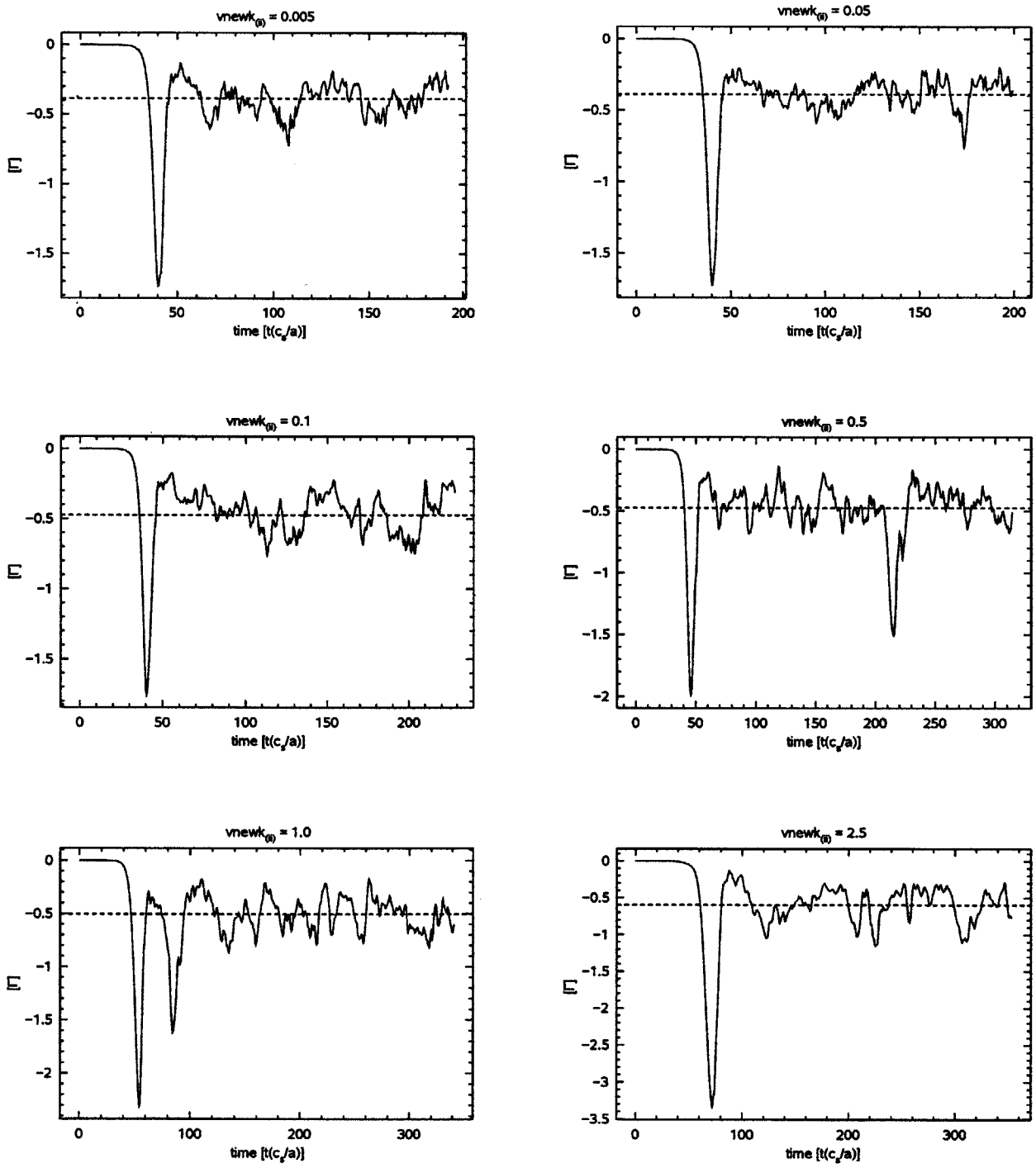


Figure 32: Electron particle flux at each probed value of vnewk.

Ion energy exchange

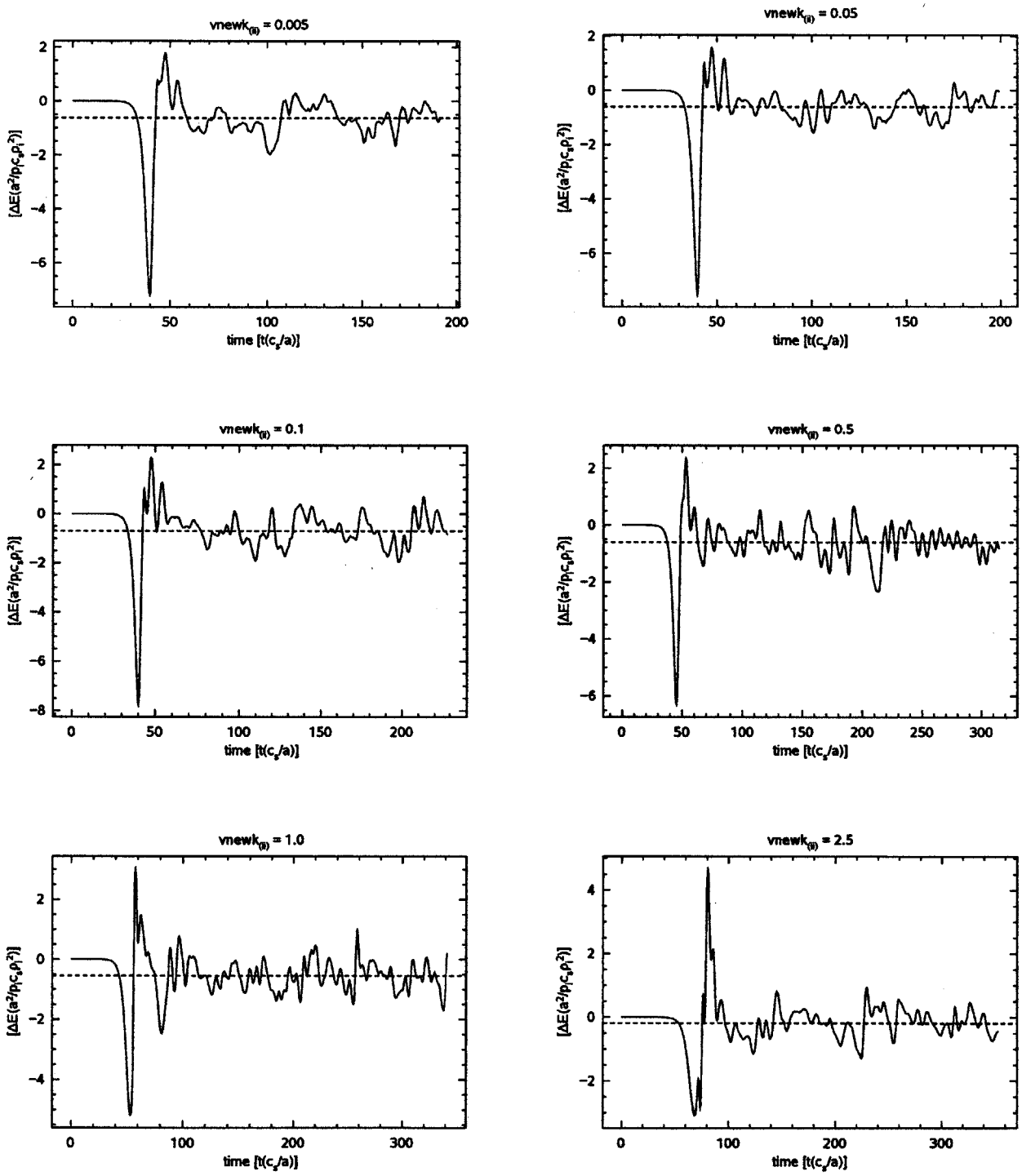


Figure 33: Ion energy exchange at each probed value of vnewk.

Electron energy exchange

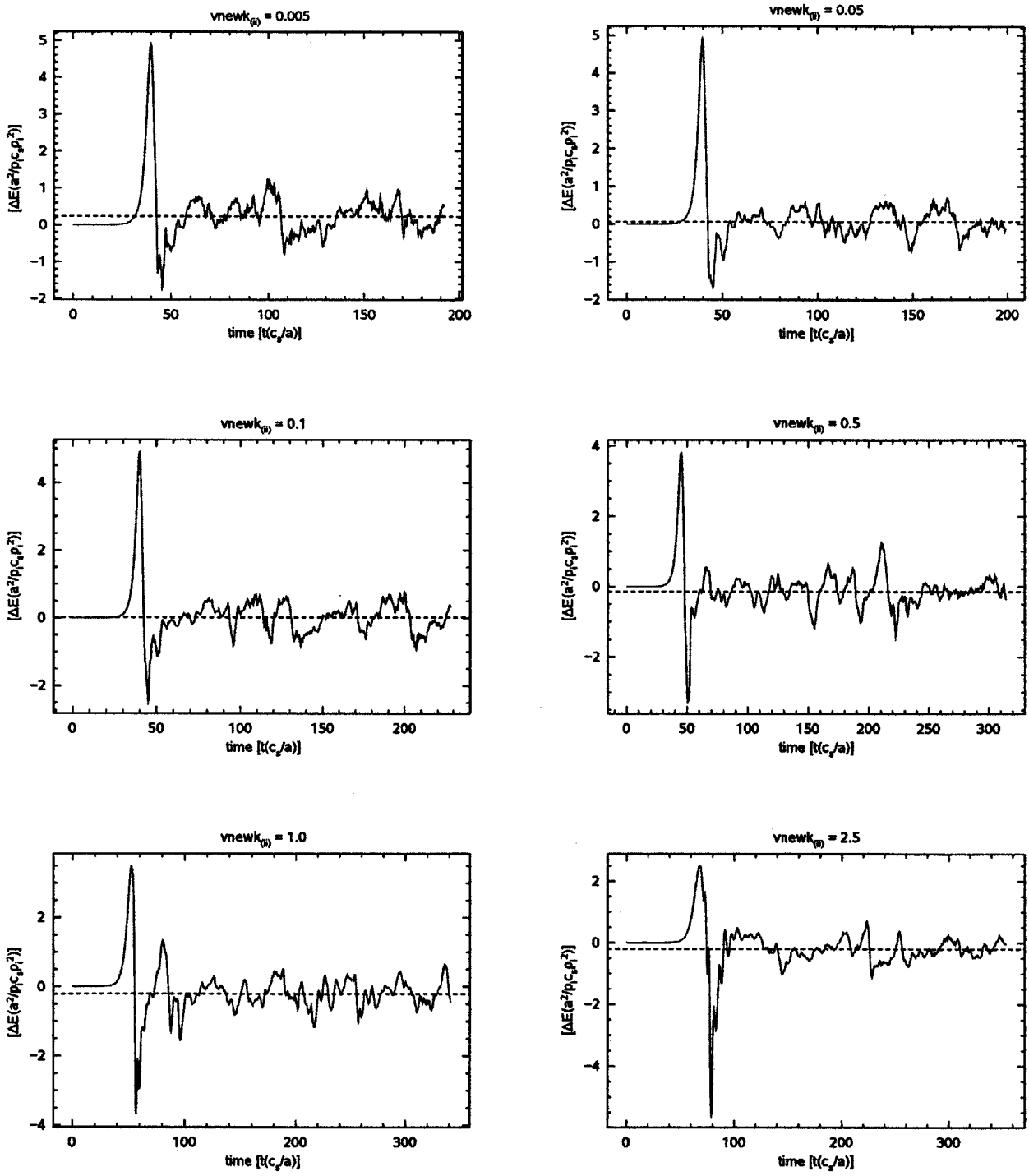


Figure 34: Electron energy exchange at each probed value of vnewk.

C Trial C time trace comparisons

Ion heat flux

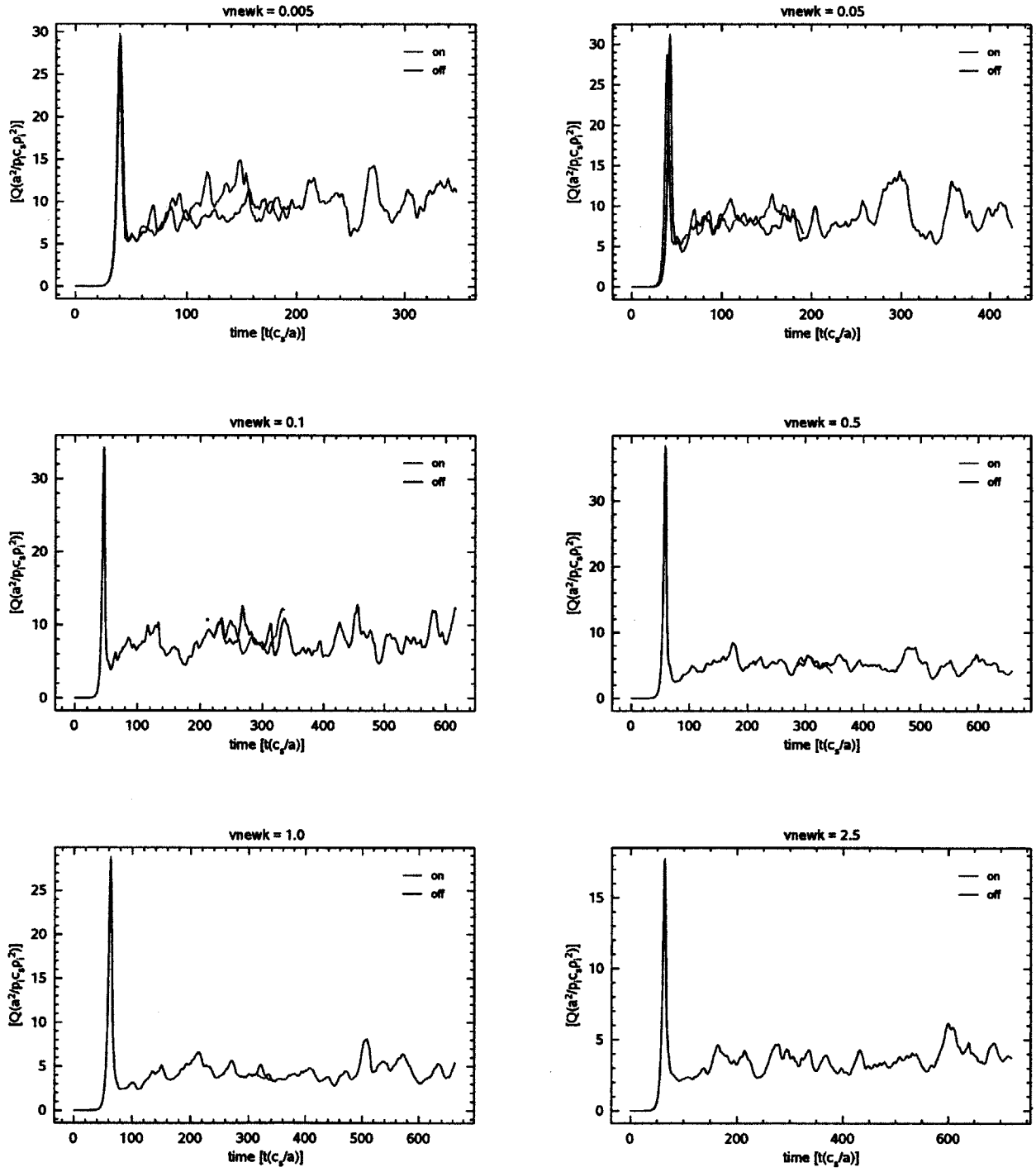


Figure 35: Turbulent ion heat flux time trace results with the collisional heating diagnostic on and off.

Electron heat flux

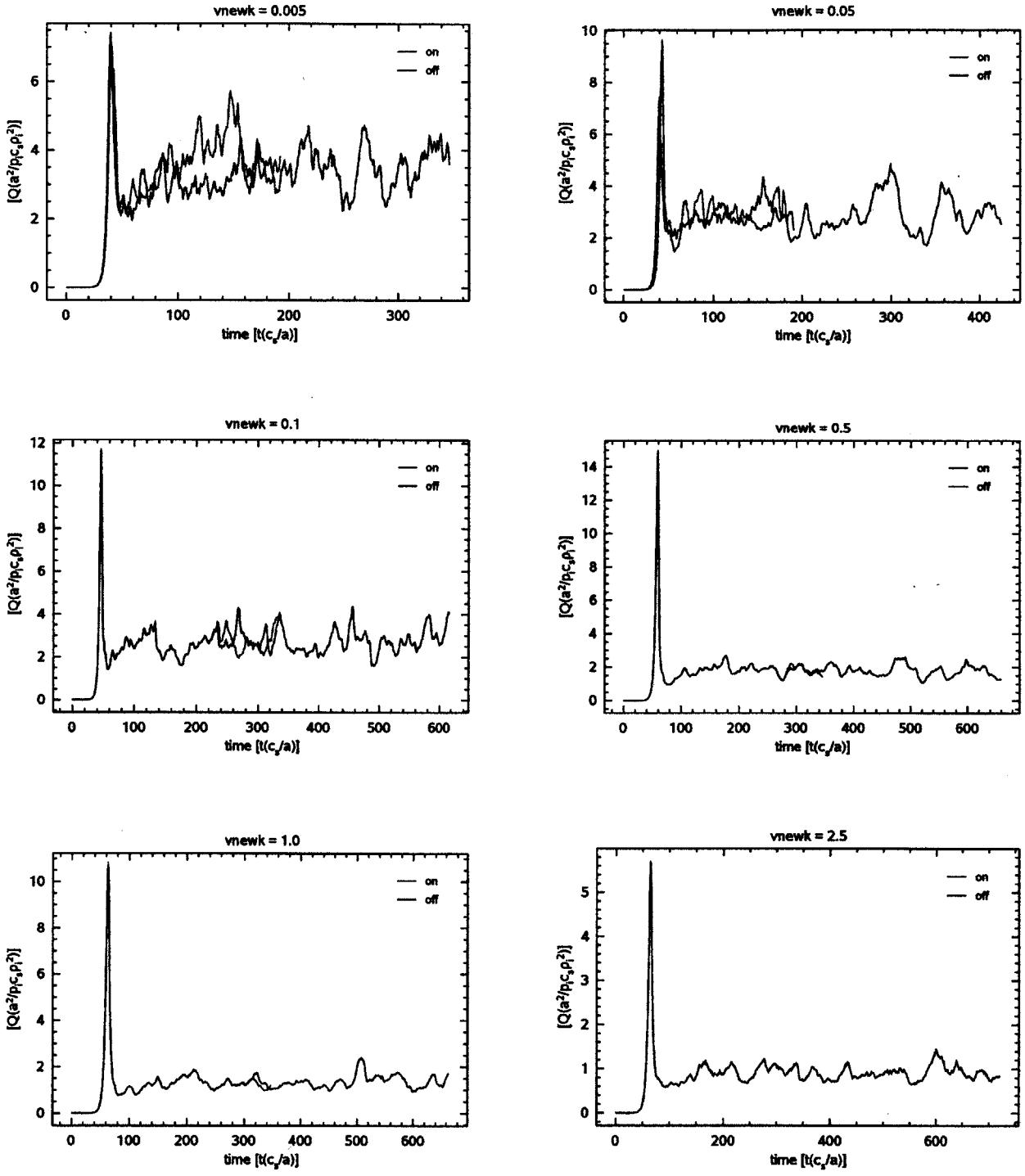


Figure 36: Turbulent electron heat flux time trace results.

Ion momentum flux

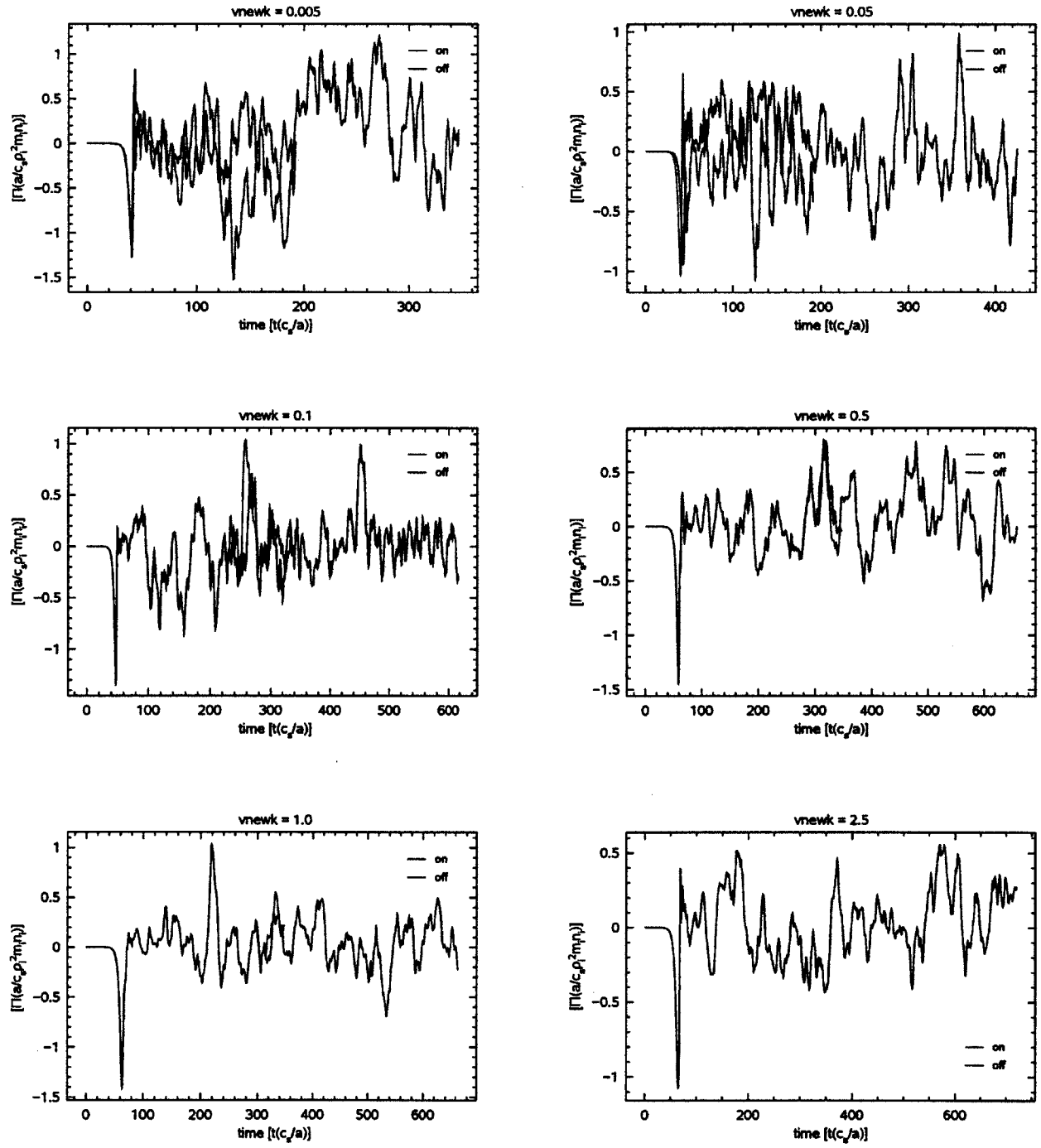


Figure 37: Turbulent ion momentum flux time trace results.

Electron momentum flux

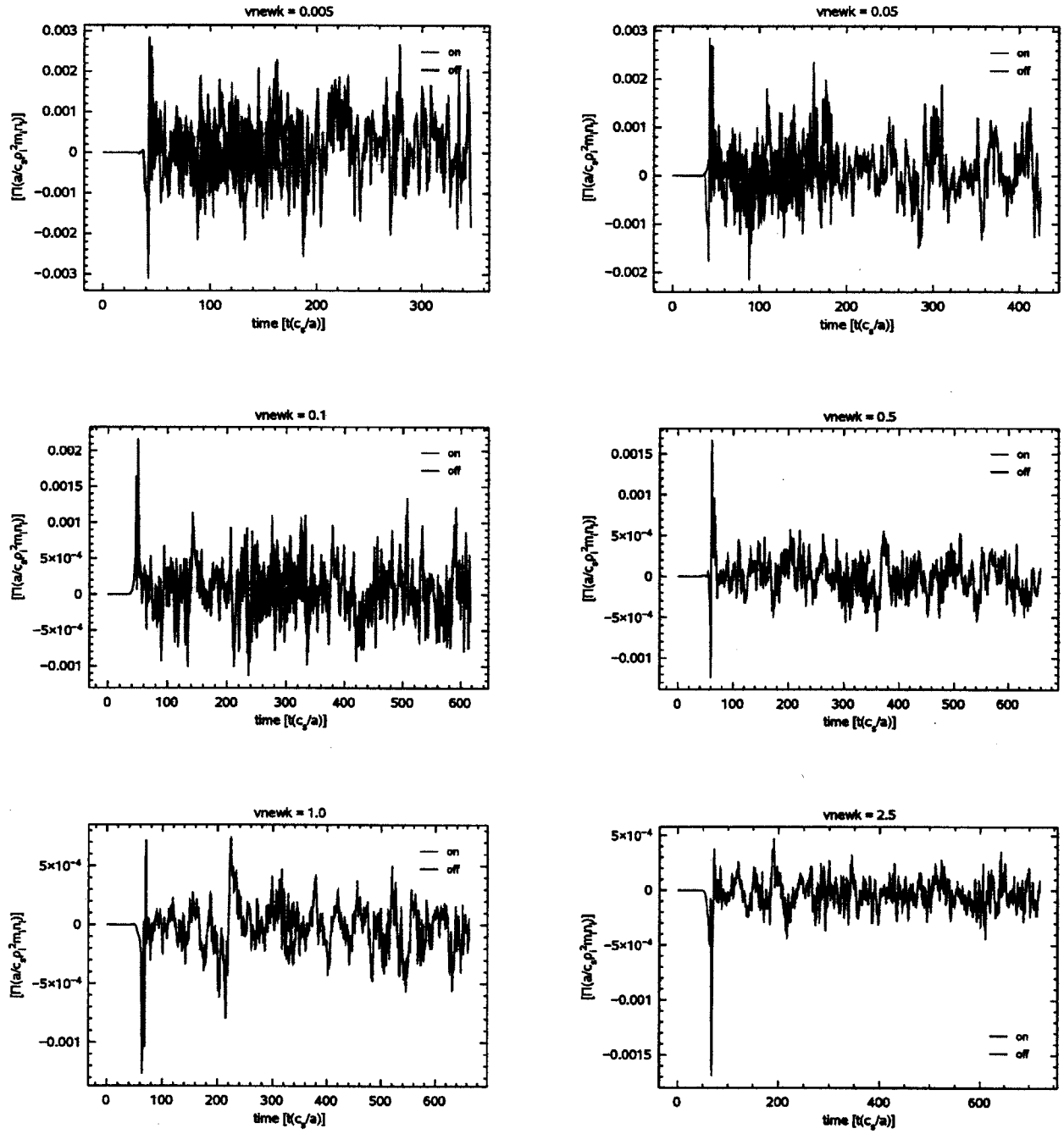


Figure 38: Turbulent electron momentum flux time trace results.

Ion particle flux

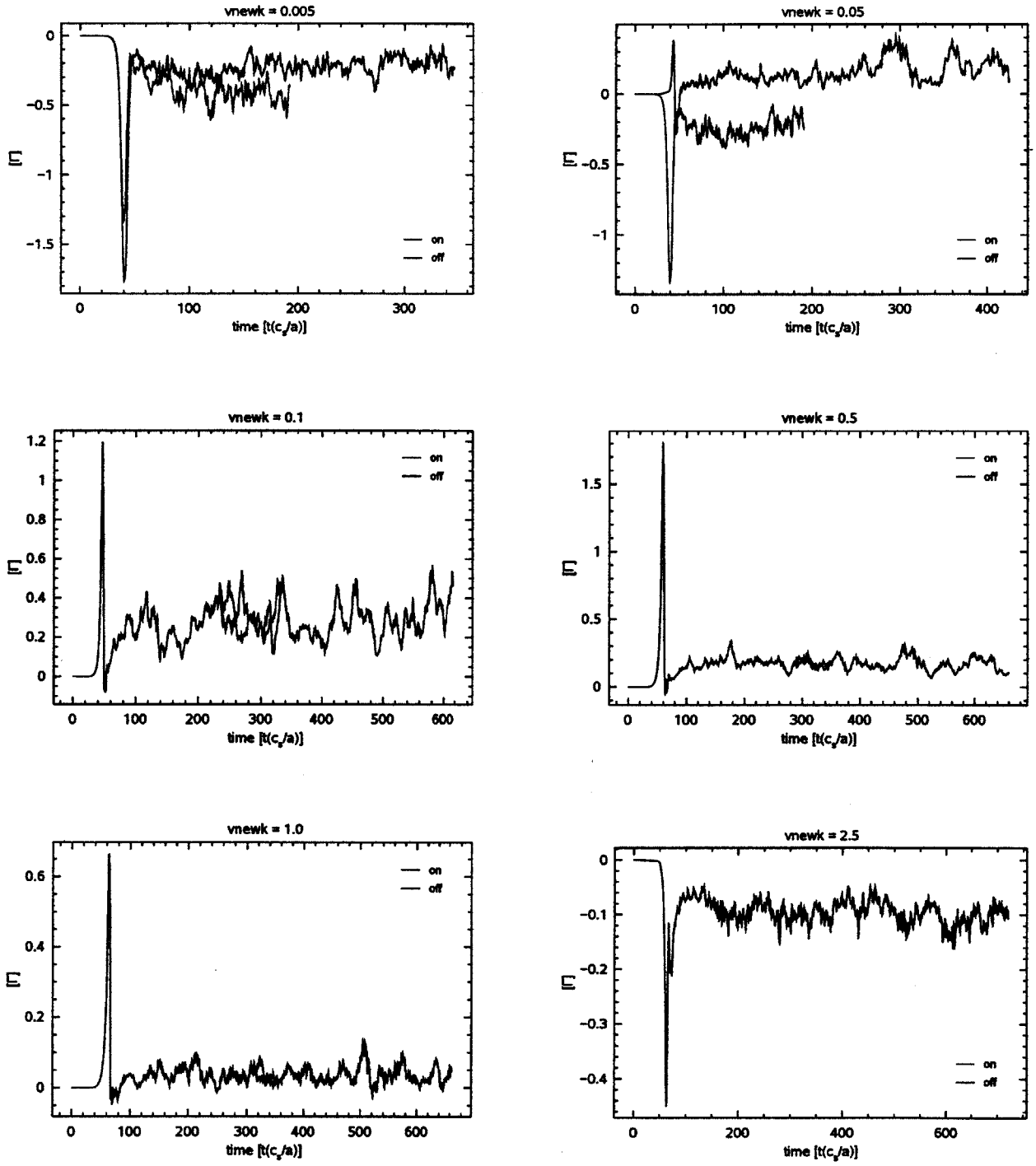


Figure 39: Turbulent ion particle flux time trace results.

Electron particle flux

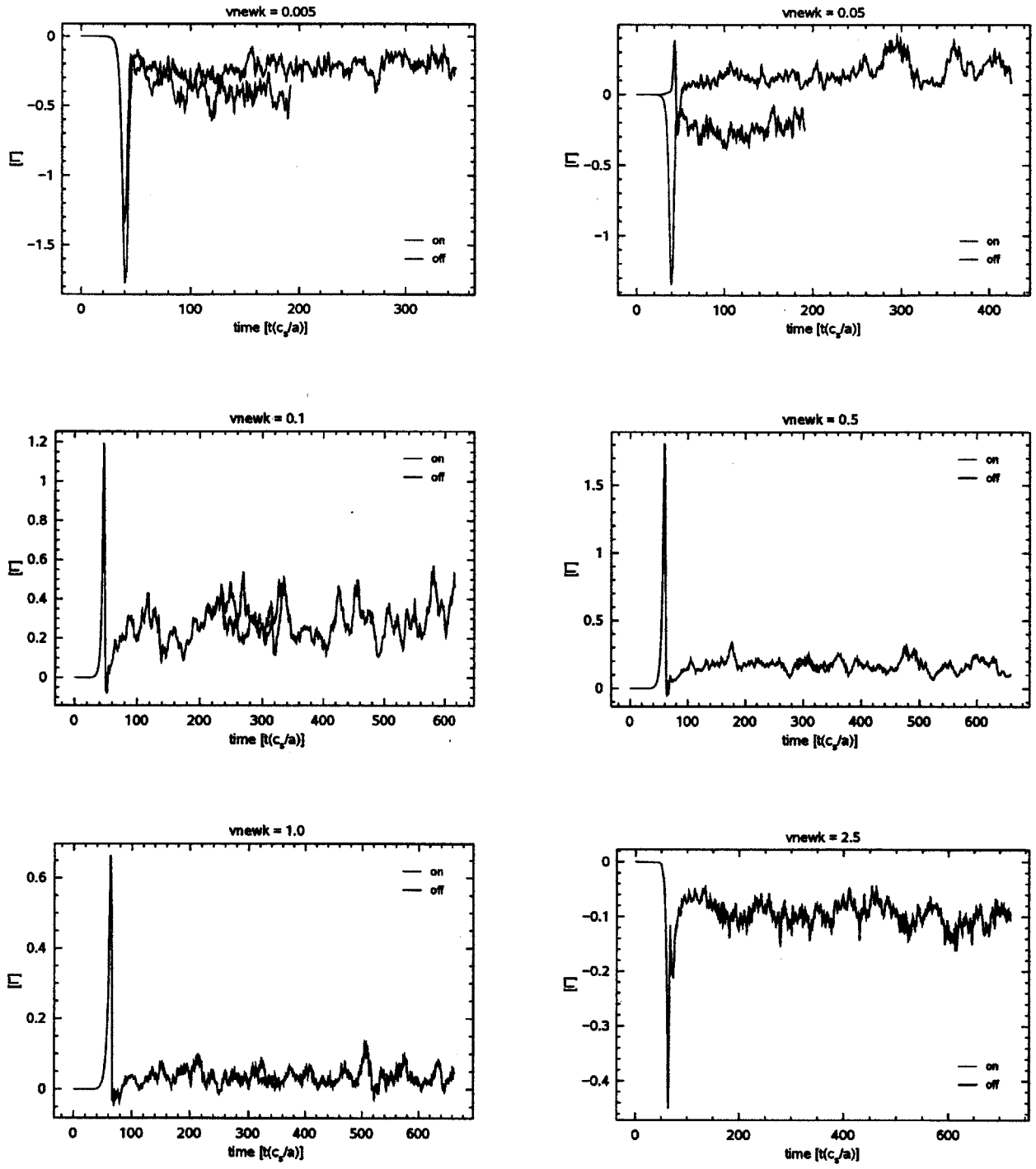


Figure 40: Turbulent electron particle flux time trace results.

Ion energy exchange

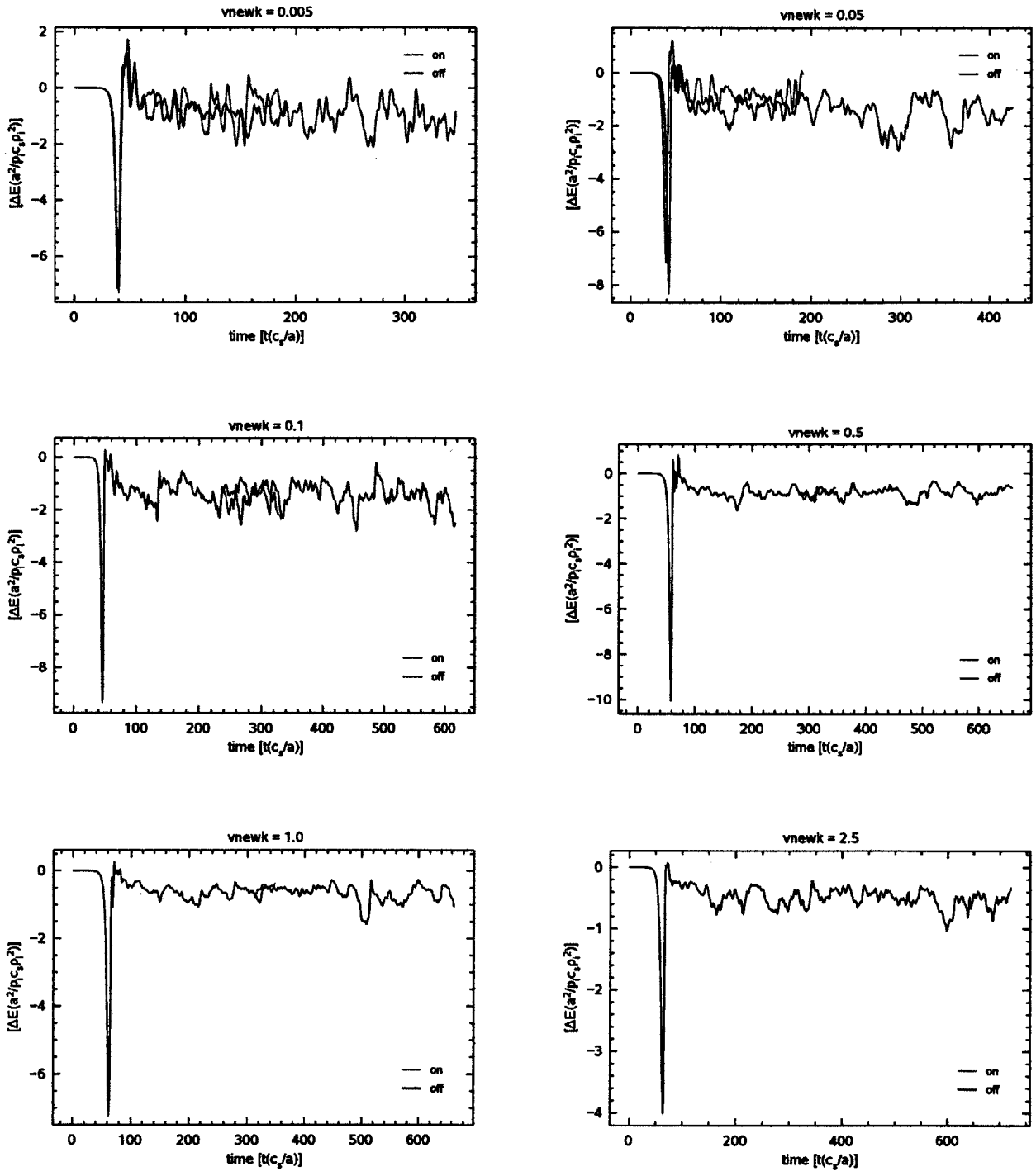


Figure 41: Turbulent ion energy exchange time trace results.

Electron energy exchange

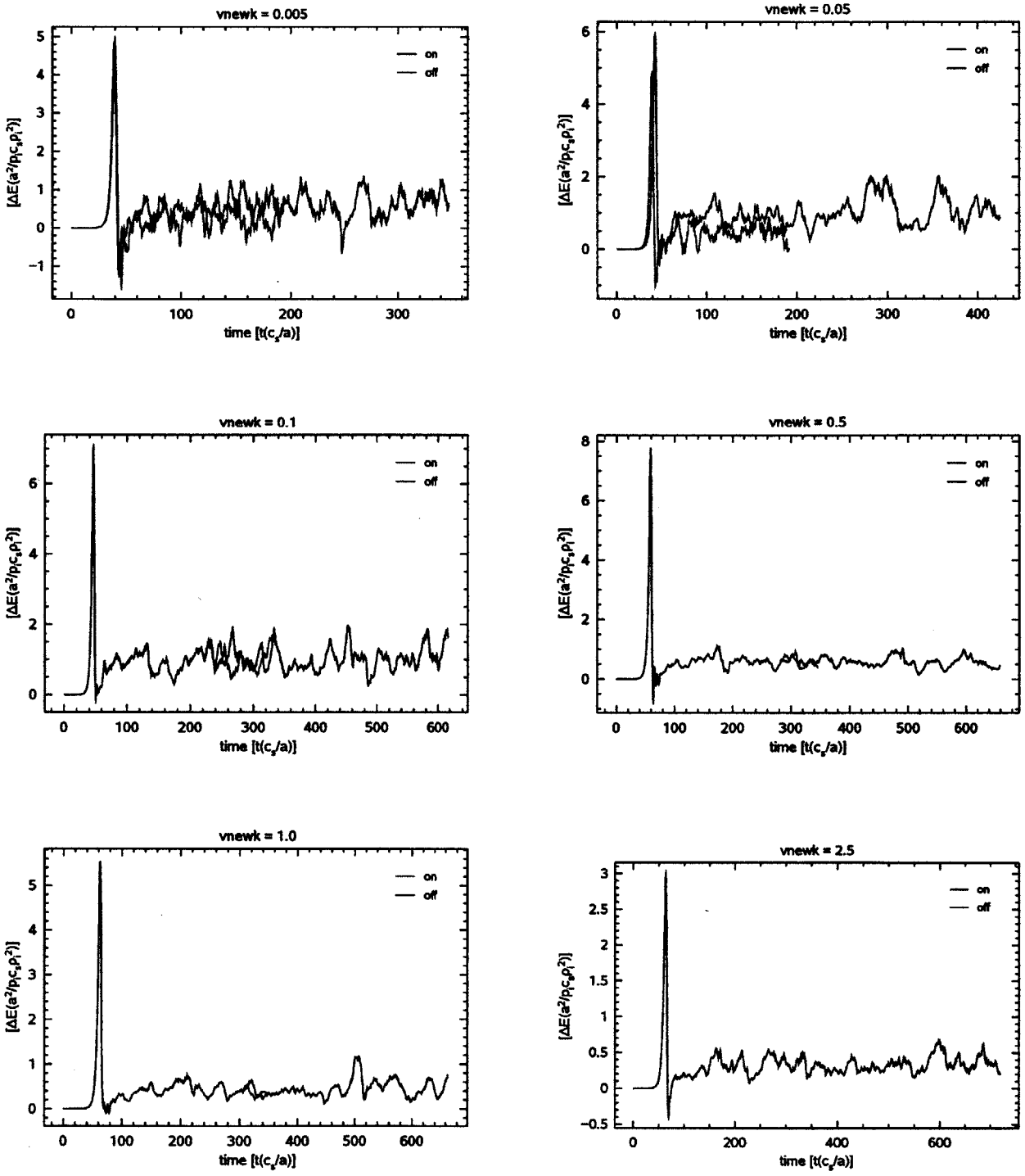


Figure 42: Turbulent energy exchange time trace results.

D Reference files

D.1 Example input file: gs.in

This input file differs slightly from the previously discussed "default" set of values (in that the electron vnewk parameter is 0.0424264 instead of 0), but serves to represent the format for specifying simulation parameters in GS2.

```
&collisions_knobs
 collision_model='default'
/

&hyper_knobs
 hyper_option = 'visc_only'
 const_amp = .false.
 isotropic_shear = .false.
 D_hypervisc = 0.05
/

&theta_grid_parameters
 ntheta= 32
 nperiod= 1

 rhoc = 0.54
 shat = 0.8
 qinp = 1.4
 Rmaj = 3.0
 R_geo = 3.0
 shift = 0.0
 akappa = 1.0
 akappri = 0.0
 tri = 0.0
 tripri = 0.0
/

&parameters
beta=  0.0
zeff=  1.0
/

&theta_grid_eik_knobs
 itor = 1
 iflux = 0
 irho = 2

 ppl_eq = F
 gen_eq = F
 efit_eq = F
 local_eq = T

 eqfile = 'dskeq.cdf'
 equal_arc = T
 bishop = 4
 s_hat_input = 0.8
 beta_prime_input = 0.0
 delrho = 1.e-3
 isym = 0
 writelots = F
/
&fields_knobs
```

```
 field_option='implicit'
/
&gs2_diagnostics_knobs
 write_symmetry = T
 write_hrate = F
 print_flux_line = T
 write_nl_flux = T
 print_line = F
 write_line = F
 write_eigenfunc = T
 write_omega = T
 write_final_fields = T
 write_g = F
 write_correlation = T
 nwrite=     50
 navg=       50
 lsave = 3000
 omegatinst = 500.0
 save_for_restart = .true.
 omegatol = -1.0e-3
/
&le_grids_knobs
 ngauss = 5
 negrid = 12
 vcut= 2.5
/
&dist_fn_knobs
 adiabatic_option="iphi00=2"
 gridfac=   1.0
 boundary_option="linked"
 !mach = -0.030342
 g_exb = 0.0533
/

&kt_grids_knobs
 grid_option='box'
/
&kt_grids_box_parameters
 ! naky = (ny-1)/3 + 1
 ny = 32
 ! nakx = 2*(nx-1)/3 + 1
 nx = 128
 ! ky_min = 1/y0
 y0 = 10.
 jtwist = 5
/
&init_g_knobs
 chop_side = F
 phiinit=   1.e-3
 restart_file = "save/restart"
 ginit_option= "noise"
/
&knobs
 fphi =   1.0
 fapar =  0.0
 faperp = 0.0
 delt = 0.05
 nstep = 50000
!neo_test = T
```

```
/

&theta_grid_knobs
 equilibrium_option='eik'
/
&nonlinear_terms_knobs
 nonlinear_mode='on'
 cfl = 0.25
/
&reinit_knobs
 delt_adj = 2.0
 delt_minimum = 1.e-4
/
&layouts_knobs
 layout = 'lxyes'
 local_field_solve = F
/

&species_knobs
nspec=           1
/
&species_parameters_1
 z=   1.0
 mass=   1.0
 dens=   1.0
 temp=   1.0
 tprim=  2.9
 fprim=  0.733
 uprim=  0.0
 vnewk=  0.000707107 ! this is 1e-3/sqrt(2)
 type='ion'
/

&dist_fn_species_knobs_1
 fexpr= 0.45
 bakdif= 0.05
/

&species_parameters_2
& z=   -1.0
& mass=   2.7e-4
& dens=   1.0
& temp=   1.0
& tprim=  2.3
& fprim=  0.733
& uprim=  0.0
& vnewk=  0.0424264
& type='electron'
&/
&dist_fn_species_knobs_2
& fexpr= 0.45
& bakdif= 0.05
&/
```

D.2 Data analysis script: `gs2_env.jl`

This script defines a `grab()` function that intakes an output file and returns all of its data as distinctly labeled arrays. It also contains a list of all output quantities typically recorded by GS2.

```
using NetCDF

varray(path,var) = ncread("$(path)/gs.out.nc", "$(var)")

# info on vars can be found at:
# http://sourceforge.net/apps/mediawiki/gyrokinetics/index.php?title=Variables

vars = [# listed by array structure

        # [t]
                "t",
                "hflux_tot",
                "phi2",
                "zflux_tot",
                "vflux_tot",
        # [kx]
                "kx",
        # [ky]
                "ky",
        # [theta]
                "theta",

        #[species, t]
                "es_energy_exchange",
                "ntot2",
                "es_part_flux",
                "es_heat_par",
                "ntot20",
                "es_heat_flux",
                "es_mom_flux",
                "es_heat_perp",

        # [kx, t]
                "phi2_by_kx",
        # [ky, t]
                "phi2_by_ky",

        # [ri, kx, t]
                "phi00",
        # [kx, ky, t]
                "phi2_by_mode",

        # [ri, kx, ky, t]
                "omegaavg",
                "omega",
                "phi0",
        # [ri, kx, species, t]
                "tpar00",
                "density00",
                "upar00",
                "ntot00",
                "tperp00",
        # [kx, ky, species, t]
                "ntot20_by_mode",
                "es_mom1_by_k",
                "es_mom0_by_k",
                "ntot2_by_mode",
                "es_perpmom_by_k",
                "tperp2_by_mode",
```

```
                "es_part_by_k",
                "es_heat_by_k",
                "es_parmom_by_k",
                "tpar2_by_mode",
                "es_mom_by_k",
        # [ri, theta, kx, ky]
                "epar",
                "phi",
                "phi_norm",
        # [ri, theta, ky, t]
                "phi_corr_2pi",

        # [theta, vpa, species, t]
                "es_mom_sym",
        # [ri, theta, kx, ky, species]
                "tpar",
                "qparflux",
                "tperp",
                "density",
                "pperpj1",
                "ntot",
                "upar"
        ]

function declare(s::String,v::Any)
        s=symbol(s)
        @eval (($s) = ($v))
end

function grab(path)
        for i = 1:length(vars)
                declare(vars[i],varray(pdath,vars[i]))
        end
end

trim(v) = reshape(v,length(v)) #cleans up arrays for easy plotting
```