Computer Physics Communications 123 (1999) 123–142

Computer Physics Communications

www.elsevier.nl/locate/cpc

# A Fortran code for solving the Kadanoff–Baym equations for a homogeneous fermion system

H.S. Köhler [a,1], N.H. Kwong [a,2], Hashim A. Yousif [b,3]

[a] *Physics Department, University of Arizona, Tucson, AZ 85721, USA*

[b] *Natural Sciences Division, University of Pittsburgh at Bradford, Bradford, PA 16701, USA*

Received 20 April 1999

**Abstract**

A Fortran code for solving the Kadanoff–Baym equations for a homogeneous fermion system is presented. These are a class of quantum kinetic equations the solutions of which are two-time Green (correlation) functions, which carry both statistical (orbital occupation distributions) and dynamical (orbital energies and widths) information about the system away from equilibrium. In this program, the system's initial state is assumed to be uncorrelated. As examples of applications, two separate situations involving equilibration (thermalization) are considered: the first one models an electron plasma in an excited semiconductor, and the other models nuclear heavy-ion collisions. The most time-consuming part of the program is the computation of the convolutions of various combinations of the Green functions. This task is efficiently performed by the use of Fast Fourier Transform. As summary output, the particle density, energy densities, and the quadrupole moment of the distribution are displayed. The program can easily be modified such that the user may use any initial distribution of fermions. 

*Keywords:* Quantum transport; Nonequilibrium Green functions; Kadanoff–Baym equations; Momentum relaxation; Semiconductor transport; Nuclear reactions

## PROGRAM SUMMARY

*Title of program:* kb

*Catalogue identifier:* ADKQ

*Program Summary URL:*
http://www.cpc.cs.qub.ac.uk/cpc/summaries/ADKQ

*Program obtainable from:* CPC Program Library, Queen's University of Belfast, N. Ireland

*Computers:* (1) Cray C90; (2) SGI

*Operating systems under which the program has been tested:* (1) UNICOS; (2) IRIX 6.4

*Programming language used:* Fortran 77

*High speed storage required:* (1) 38 Mwords; (2) 148 Mwords (for ntl = 61, nq = 4620, idim = 21, nvec = 48, and mod_store = 3, see Table 3)

[1] E-mail: Kohler@physics.arizona.edu

[2] E-mail: Kwong@physics.arizona.edu

[3] E-mail: yousif@vms.cis.pitt.edu

0010-4655/99/$ – see front matter 
PII: S0010-4655(99)00260-X

*No. of bits in a word:* (1) 64; (2) 32

*Peripherals used:* Printer, terminal

*No. of bytes in distributed program, including test data, etc.:* 35862

*Distribution format:* ASCII

*External routines:* The program uses the three-dimensional Fast Fourier Transform routine CFFT3D of the SGI library or the subroutine CCFFT3D of the Cray library

*Keywords:* Quantum transport, nonequilibrium Green functions, Kadanoff–Baym equations, momentum relaxation, semiconductor transport, nuclear reactions

*Nature of physical problem*
The output of this program describes the thermalisation by internal collisions in a nonrelativistic infinite fermion system. It is most useful when applied to situations where the time scales of relaxation, individual collisions, and quantum coherence are overlapping: some examples are nuclear heavy ion collisions and femtosecond-scale carrier kinetics in excited semiconductors.

*Method of solution*
The equations are solved on a momentum grid, marching along both axes in the two-time plane. The bulk of the processing time is spent in the time-stepping and evaluating the collision integrals. The time-stepping is done by a two-pass iteration, while the collision integrals are computed by Fast Fourier Transform.

*Typical running time*
The running time depends mainly on the parameter ntimes. Table 1 shows the running times on the Cray C90 of the Pittsburgh Supercomputing Center and on the SGI of the University of Arizona Center of Computing and Information Technology. The CRAY C90 has 16 processors with a peak aggregate speed of 16 Gflops, and a main memory of 512 Mwords or 4 Gbytes. The calculations of Table 1 were made for the nuclear case and for the following parameters: ntl = 61, nq = 4620, nvec = 48, idim = 21, and mod_store = 3 (cylindrical symmetry). Serial processing on one processor was used on the SGI, and parallel processing was used on the CRAY C90.

Table 1
CPU in seconds

| ntimes | CPU on CRAY C90 (s) | CPU on SGI (s) |
|---|---|---|
| 10 | 54 | 92 |
| 30 | 356 | 783 |
| 60 | 1312 | 3853 |

# LONG WRITE-UP

## 1. Introduction

Two-time Green functions play a key role in the theory of nonequilibrium statistical mechanics of quantum many-body systems. Their general properties, perturbation theory, and equations of motion have been expounded upon in classic articles and monographs [1–5]. Notably, this formalism clarifies the approximations that are to be made to obtain the Boltzmann equation in the weak correlation limit. It also allows the derivation of generalized transport equations applicable to strongly correlated systems. Recently, the two-time Green functions have found extensive applications in two areas where the Boltzmann assumptions are not valid: short-time-scale semiconductor optics (e.g. [6] and references therein, [7–9]), intermediate-energy nuclear collisions (e.g. [10–12]), and pion production [13]. These investigations have yielded considerable understanding of such non-Boltzmann features as finite memory-time of correlations and off-energy-shell propagation and decay of (quasi-)particles in a medium. Applications to other areas of quantum transport such as quantum fields have also been discussed [5,14,15].

In this paper, we present a Fortran program that solves the equations of motion of the two-time Green functions for a homogeneous fermion system. The version of the equations that we use, usually known as the Kadanoff–Baym (KB) equations [2], are a set of coupled integro-differential equations in time. The integral kernel, called the self-energy, is in general too complicated to calculate in the exact theory. So all known applications start with approximating the self-energy as a functional of the Green functions in a way which is equivalent to summing a selected subset of terms of its perturbation expansion. One class of approximations, called *conserving* by Kadanoff

and Baym [2], are especially important because they preserve the conservation laws for number, momentum, and (kinetic + potential) energy if the exact theory obeys these laws. The simplest of such approximations, which allows relaxation in the solution to the KB equations, is the direct-Born approximation (see Section 2). The present code solves the KB equations within this approximation. Two specific applications are included here: (i) the relaxation of charge carriers in an excited semiconductor, and (ii) the thermalization of two counter-streaming slabs of nuclear matter, which models a heavy-ion collision. As typical output, particle and energy densities and the quadrupole moment of the momentum distribution as functions of time are displayed. The code has been extended to include phonons [16] and a two-band model of a semiconductor [9]. Work to include mesonic degrees of freedom in nuclear matter is also under way.

Except for the pioneering work of [10], the direct numerical solution of the KB equations [7–9,11,13] (with a specified self-energy function) has been undertaken only recently. In most studies using the two-time Green function framework, further approximations are taken to reduce the KB equations to simpler kinetic equations before they are analyzed and/or solved. Discussions of the relative merits of the two approaches may be found in the references. We believe, with the increasing capacity of available computing resources, the direct numerical approach will gain importance.

## 2. Mathematical expressions

The Hamiltonian considered here consists of the kinetic energy and the sum of pairwise-interacting potential energy terms. The two-body potential is assumed to be translationally invariant so that the system's total momentum is conserved. We consider only bulk, spatially homogeneous normal fermion systems [4] .

The central dynamical variables in this approach, the two-time Green (correlation) functions, are defined as

$$G^{<}(\boldsymbol{p},t_1,t_2) = i\langle a^{\dagger}_{\boldsymbol{p}}(t_2)a_{\boldsymbol{p}}(t_1)\rangle,$$
$$G^{>}(\boldsymbol{p},t_1,t_2) = -i\langle a_{\boldsymbol{p}}(t_1)a^{\dagger}_{\boldsymbol{p}}(t_2)\rangle,$$

where $a_{\boldsymbol{p}}$ ($a^{\dagger}_{\boldsymbol{p}}$) are the Heisenberg-picture annihilation (creation) operators for a particle of momentum $\boldsymbol{p}$ (in this paper, the momentum variable $\boldsymbol{p}$ is defined as the actual momentum divided by $\hbar$) and $\langle\cdots\rangle$ denotes averaging over a density matrix representing the initial state of the system. As explained below, the $G^{\gtrless}$ contain all one-particle information about the evolving system. Their equations of motion, the KB equations, are

$$\left[i\hbar\frac{\partial}{\partial t_1} - \frac{\hbar^2p^2}{2m}\right]G^{\gtrless}(\boldsymbol{p},t_1,t_2) = I_1^{\gtrless}(\boldsymbol{p},t_1,t_2),$$
$$\left[-i\hbar\frac{\partial}{\partial t_2} - \frac{\hbar^2p^2}{2m}\right]G^{\gtrless}(\boldsymbol{p},t_1,t_2) = I_2^{\gtrless}(\boldsymbol{p},t_1,t_2). \tag{1}$$

The right-hand side of each equation consists of a Hartree–Fock potential term and a collision integral,

$$I_1^{\gtrless}(\boldsymbol{p},t_1,t_2) = \Sigma^{\mathrm{HF}}(\boldsymbol{p},t_1)G^{\gtrless}(\boldsymbol{p},t_1,t_2) + \int_{t_0}^{t_1}\frac{d\bar{t}}{\hbar}\big[\Sigma^{>}(\boldsymbol{p},t_1,\bar{t}) - \Sigma^{<}(\boldsymbol{p},t_1,\bar{t})\big]G^{\gtrless}(\boldsymbol{p},\bar{t},t_2)$$
$$- \int_{t_0}^{t_2}\frac{d\bar{t}}{\hbar}\Sigma^{\gtrless}(\boldsymbol{p},t_1,\bar{t})\big[G^{>}(\boldsymbol{p},\bar{t},t_2) - G^{<}(\boldsymbol{p},\bar{t},t_2)\big],$$

[4] The possibility of spontaneous symmetry breakings, or of nonvanishing anomalous Green functions, is not included here.

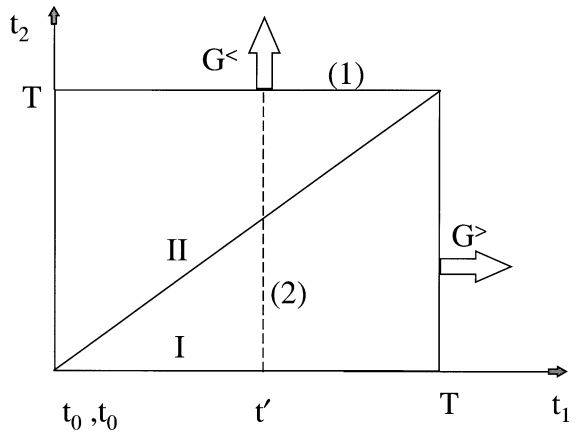


Fig. 1. To time step $G^<$ along the $t_2$ axis from $T$ to $T+\Delta t$ at $t_1=t'$ one needs $\Sigma^<$ and $\Sigma^>$ along line (1). One also needs $G^<$ and $G^<$ along the broken line (2).

$$I_2^{\gtrless}(\boldsymbol{p},t_1,t_2)=G^{\gtrless}(\boldsymbol{p},t_1,t_2)\Sigma^{\mathrm{HF}}(\boldsymbol{p},t_2)+\int_{t_0}^{t_1}\frac{d\bar{t}}{\hbar}\big[G^>(\boldsymbol{p},t_1,\bar{t})-G^<(\boldsymbol{p},t_1,\bar{t})\big]\Sigma^{\gtrless}(\boldsymbol{p},\bar{t},t_2)$$
$$-\int_{t_0}^{t_2}\frac{d\bar{t}}{\hbar}G^{\gtrless}(\boldsymbol{p},t_1,\bar{t})\big[\Sigma^>(\boldsymbol{p},\bar{t},t_2)-\Sigma^<(\boldsymbol{p},\bar{t},t_2)\big]. \tag{2}$$

The Hartree–Fock energy is given by

$$\Sigma^{\mathrm{HF}}(\boldsymbol{p},t)=-i\left[\nu V(0)\int\frac{d^3p_1}{(2\pi)^3}G^<(\boldsymbol{p}_1,t,t)-\int\frac{d^3p_1}{(2\pi)^3}G^<(\boldsymbol{p}_1,t,t)V(\boldsymbol{p}-\boldsymbol{p}_1)\right], \tag{3}$$

where $\nu$ is the degeneracy factor: $\nu=2,4$ for electrons and nucleons, respectively. The correlation self-energy $\Sigma^{\gtrless}$ is given in the direct-Born approximation as

$$\Sigma^{\gtrless}(\boldsymbol{p},t_1,t_2)=\nu\int\frac{d^3p_1}{(2\pi)^3}G^{\gtrless}(\boldsymbol{p}_1,t_1,t_2)\big[V(\boldsymbol{p}-\boldsymbol{p}_1)\big]^2$$
$$\times\int\frac{d^3p_2}{(2\pi)^3}G^{\gtrless}(\boldsymbol{p}-\boldsymbol{p}_1+\boldsymbol{p}_2,t_1,t_2)G^{\lessgtr}(\boldsymbol{p}_2,t_2,t_1). \tag{4}$$

In practice, effective two-body potentials are sometimes used (in particular for the strongly interacting nuclear system) in calculating $\Sigma^{\mathrm{HF}}$ and $\Sigma^{\gtrless}$ to take into account effects of higher-order correlations not included in the direct-Born approximation. For the same model system, the effective potential used in $\Sigma^{\mathrm{HF}}$ may be different from that used in $\Sigma^{\gtrless}$. The choice of potentials is stated and discussed in Sections 3.3.10–3.3.13. Furthermore, the present code only treats local spin(isospin)-independent potentials, i.e., potentials which are functions of only the momentum transfer.

For each calculation, $G^{\gtrless}(\boldsymbol{p},t_0,t_0)$ is given at the initial time-point $t_1=t_2=t_0$, and the evolution proceeds along both time coordinate axes $t_1$ and $t_2$, which is graphically represented in Fig. 1. This boundary condition implies the assumption that the initial state is uncorrelated. For the treatment of initially correlated states see Refs. [10,11]. Eq. (1) is solved on a Cartesian grid in $\boldsymbol{p}$-space. The momentum integrals in Eqs. (3) and (4) are approximated by the sum of the functional values of the integrands on the grid points multiplied by the grid cell volume. For a

specific application, the time-step size and the grid-point separation are progressively reduced and the total grid-box volume increased until convergence of the physical results has been achieved to a desired accuracy.

The physical meanings of some components of $G^{\gtrless}$ follow from their definitions [2]. For example, along the time diagonal in Fig. 1, i.e., $t_1 = t_2 = t$, $G^<$ is the one-particle momentum distribution function,

$$f(\boldsymbol{p}, t) = -iG^<(\boldsymbol{p}, t, t). \tag{5}$$

At large times, when the system has reached equilibrium, one can obtain the one-particle spectral function [2] from the off-time-diagonal part of $G^{\gtrless}$,

$$A(\boldsymbol{p}, t_1, t_2) = i\big[G^>(\boldsymbol{p}, t_1, t_2) - G^<(\boldsymbol{p}, t_1, t_2)\big]. \tag{6}$$

Before the system equilibrates, this quantity still yields spectral information on the propagation of a particle in the evolving medium.

The Green functions, collision integrals, and the self-energy obey the following symmetries:

$$F^{\gtrless}(\boldsymbol{p}, t_1, t_2) = -\big[F^{\gtrless}(\boldsymbol{p}, t_2, t_1)\big]^*, \tag{7}$$

where $F$ is $G$ or $\Sigma$ and $^*$ denotes complex conjugation. Also,

$$I_1^{\gtrless}(\boldsymbol{p}, t_1, t_2) = -\big[I_2^{\gtrless}(\boldsymbol{p}, t_2, t_1)\big]^*. \tag{8}$$

In addition, for equal times ($t_1 = t_2 = t$), the following relations are satisfied:

$$G^>(\boldsymbol{p}, t, t) = -i + G^<(\boldsymbol{p}, t, t), \tag{9}$$

$$I_1^>(\boldsymbol{p}, t, t) = I_1^<(\boldsymbol{p}, t, t), \tag{10}$$

$$I_2^>(\boldsymbol{p}, t, t) = I_2^<(\boldsymbol{p}, t, t). \tag{11}$$

These relations are used in the code to reduce memory storage and runtime.

## 3. The program

The computer program consists of three parts: the input file, the include statements file, and the main section. Below is a brief description of each section.

### 3.1. The input file

The main section of the program reads the input data from this file. Table 1 shows a brief description of each of the input parameter. The following unit conventions for the input/output are adopted. In the semiconductor case, the units for momenta, energies, and time are the reciprocal exciton Bohr radius ($a_B^{-1}$), the exciton Rydberg (Ry), and femtosecond (fs), respectively[5]. In the nuclear case, the respective units are inverse fermi ($\mathrm{fm}^{-1}$), mega-electron-volt (MeV), and fermi divided by the vacuum speed of light (fm/c). Some quantities are rescaled to other units in the computing parts of the main program, but they are converted back to the units given here before being outputted.

### 3.2. The include statements file

This file contains dimension parameters and include statements. There are four parameters which the user needs to specify. Descriptions of these parameters are given in Table 3.

[5] The values of the Bohr radius and the Rydberg need to be supplied in the input file (see Table 2).

Table 2
Input data

| Program variable | Meaning |
|---|---|
| isys | = 1: the system is electrons; 2: the system is nucleons |
| aindb | Bohr radius ($a_B$) of exciton in angstroms (Å) |
| eryd | one exciton Rydberg (Ry) in eV |
| amee | ratio of the electron effective mass to the bare mass |
| epsb | background dielectric constant ($\epsilon_b$) |
| isck | 1: screening length calculated self-consistently with distribution |
| akp | inverse screening length in units of ($a_B^{-1}$) |
| ra,rhb,rgam,rgam1,rgam2 | semiconductor initial distribution parameters (see Table 4) |
| p0,pf | nuclear initial distribution parameters (see Table 5) |
| dpz | grid size in ($a_B^{-1}$) for electrons and ($\mathrm{fm}^{-1}$) for nucleons |
| dft | time-step size in fs for electrons and fm/c for nucleons |
| ntimes | number of time steps in the calculation |
| time0 | initial time |
| istf | = 0: one pass per time step; 1: two passes per time step |
| mod_store | storage mode (see Section 3.3.1) |
| ifk | = 0: mean field omitted; 1: mean field included |
| nplot | number of times when distributions are outputted (see Section 3.3.15) |
| iplot | array of time points (in units of time steps) when distributions are outputted |

Table 3
Parameters of subroutine include.f

| Parameter | Meaning |
|---|---|
| ntl | maximum number of time steps allowed |
| nq | number of nonequivalent grid points (see Section 3.3.1) |
| idim | size of the momentum grid on one side of zero in each dimension |
| nvec | size of array (in each spatial dimension) used in the FFT subroutines |

### *3.3. The main section*

The main section of the program consists of 15 subroutines. These subroutines perform various functions as we will discuss in the following subsections. The main section of the program exchanges information with the subroutines mainly by common blocks.

#### *3.3.1. Subroutine INIT_GRID*

To take advantage of Fast Fourier Transform, the KB equations are solved on a Cartesian grid in momentum space. Geometrical symmetries of the problem, when they exist, introduce redundancy among the grid points which can be exploited to reduce processing time and storage requirements. In our code, for a certain symmetry, the grid points are grouped into equivalence classes such that the solution (Green functions) has the same value at

all points within a class. The classes are ordered and each labeled by a positive integer. The Green functions and self energies are stored as arrays indexed linearly by the equivalence class labels.

The subroutine INIT_GRID maps the Cartesian grid points to the equivalence classes and vice versa for several simple symmetries. Its input is the argument *mod_store*, and its output are the arrays $irq, ix, jx, kx$.

The array $irq(i, j, k)$ gives the label of the equivalence class to which the Cartesian grid point $(i, j, k)$ belongs, where $i, j, k$ satisfy $i^2 + j^2 + k^2 \leqslant idim^2$, i.e., all functions are defined within a sphere of radius *idim* but are assumed to be zero outside this sphere. (See also below regarding the FFT routine, aliasing and zero-padding.) The arrays $ix(iq), jx(iq), kx(iq), iq = 1, nq$ give the coordinates $(i, j, k)$ of a representative Cartesian point in the class labeled $iq$, with $nq$ being the total number of classes of a particular symmetry. The program allows the user to choose from several symmetry options discussed below. The options are controlled by the parameter *mod_store* which is passed by the argument of this routine.

If *mod_store* is zero, no symmetry is assumed and the subroutine invokes a full storage mode. The subroutine assigns a class of labels to each Cartesian point within the sphere and stores the labels in the array *irq*.

If *mod_store* is one, reflection symmetry about all three axes is assumed. Points in the first octant are chosen as representative points of the equivalence classes.

If *mod_store* is two, spherical symmetry is assumed. For each integer *ip*2 between 0 and $idim^2$, the subroutine goes through the grid and counts the number of points having $i^2 + j^2 + k^2 = ip2$ and stores the numbers in the one-dimensional array *nfreq*. Thus each *ip*2 with a nonzero *nfreq* corresponds to an equivalence class. All such values of *ip*2 are then ordered and labeled. The labels are stored in the array *labelq*, from which the array *irq* is defined.

If *mod_store* is three, cylindrical symmetry is assumed. The array *irq* is defined in the same way as the previous case except that here, for each $k$-coordinate value, the subroutine counts the number of grid points having $i^2 + j^2$ equal to a certain integer.

For the last three cases, the symmetries are used to assign labels to the points in other octants or regions.

### 3.3.2. Subroutine INIT_VAR

This subroutine initializes arrays and variables.

### 3.3.3. Subroutine INIT_ELEC

This subroutine and the following one set up initial distributions for the test runs included in this paper (Section 6). The user can of course choose some other distributions. As an illustrative example for the semiconductor calculations, we have chosen a momentum distribution which has been used to model the electron distribution in the conduction band of GaAs immediately after excitation by irradiation by a strong, short-pulsed (10–100 fs) laser [7]. The expression is

$$f(\boldsymbol{p}, 0) = \frac{A}{2}\big[3 f_h(p)(1 - \cos^2\theta) + f_l(p)(1 + 3\cos^2\theta)\big], \tag{12}$$

where $(p, \theta, \phi)$ are the polar coordinates of the vector $\boldsymbol{p}$, and

$$f_h(p) = \mathrm{e}^{-(\chi_0 - \chi_e(p) - \chi_h(p))^2/2\gamma},$$
$$f_l(p) = \mathrm{e}^{-(\chi_0 - \chi_e(p) - \chi_l(p))^2/2\gamma},$$
$$\chi_e(p) = \frac{\hbar^2 p^2}{2m},$$
$$\chi_h(p) = \frac{\hbar^2}{2m_0}(\gamma_1 - 2\gamma_2)p^2,$$
$$\chi_l(p) = \frac{\hbar^2}{2m_0}(\gamma_1 + 2\gamma_2)p^2,$$
$$m = m_0 m^*.$$

Table 4
Variables of subroutine INIT_ELEC

| Program variables | Mathematical symbols | Comments |
|---|---|---|
| amee | $m^* = m/m_0$ | ratio of the electron effective mass to the free mass |
| anu | $\nu$ | degeneracy factor = 2 |
| hbar | $\hbar = .658212$ eV-fs | Planck's constant |
| h2m2 | $\hbar^2/2m$ | $\frac{\hbar^2}{2m_0} = 3.80998$ eV Å$^2$ |
| ra | $A$ | A parameter, see Eq. (12) |
| rgam | $\gamma$ | a parameter, see Eq. (12) |
| rgam1 | $\gamma_1$ | a parameter, see Eq. (12) |
| rgam2 | $\gamma_2$ | a parameter, see Eq. (12) |
| rhb | $\chi_0$ | A parameter, see Eq. (12) |
| coul | $\frac{e^2}{\epsilon_b}$ | $e^2 = 14.3997$ eV-Å, $e$ is the electronic charge |

The subscripts $h$ and $l$, standing for 'heavy-hole band' and 'light-hole-band', respectively, indicate the contributions of the two valence bands from which the electrons are excited. $m_0$ is the free electron mass, while $m$ is the effective mass of the electron in the conduction band. $\chi_e$, $\chi_h$, $\chi_l$ are the kinetic energies of electrons or holes in the respective bands. $\chi_0$ is the excess of the center energy of the laser over the band gap; it is always tuned well into the conduction band. The difference in the mass parameters of the two valence bands (when $\gamma_2 \neq 0$) introduces an anisotropy in the distribution (12). Table 4 contains the program names of the constants $A$, $\chi_0$, $\gamma$, $\gamma_1$, $\gamma_2$, $m_0$, and $m^*$. Their values are set in the input data file.

The subroutines POTCOUL and POTCOUL_FK are called to set up the potentials used in $\Sigma^{\gtrless}$ and $\Sigma^{\mathrm{HF}}$. If a self-consistent screening constant *kappa* is to be calculated, the subroutine CALAKP is also called.

Treating a one-band model, the present program can only be used to investigate the incoherent kinetics of the electron plasma. It obviously cannot describe processes, such as excitons, that are coherent with the laser field. For this purpose a two-band extension of the program is needed [9].

### *3.3.4. Subroutine INIT_NUCL*

This subroutine sets up a nuclear initial distribution. This distribution is defined by the following expression:

$$f(\boldsymbol{p}, 0) = \frac{1}{\mathrm{e}^{-(1/k_B T)(\hbar^2 \bar{p}^2/2m - \mu)} + 1}, \tag{13}$$

where

$$\bar{p} = \sqrt{p_x^2 + p_y^2 + (|p_z| - p_0)^2}. \tag{14}$$

At zero temperature, Eq. (13) reduces to

$$\begin{aligned} f(\boldsymbol{p}, 0) &= 1 \quad \text{for } p_f - \bar{p} > 0, \\ &= 0 \quad \text{for } p_f - \bar{p} \leqslant 0, \end{aligned} \tag{15}$$

where $p_f$ is the Fermi momentum. This distribution describes two counter-streaming slabs of nuclear matter boosted relative to each other by $p_0$ [10,11]. It models the interior of the interaction region during the initial stage of a central collision between two heavy ions. Table 5 contains the variables and constants defined in this subroutine.

The subroutines POTPAW and POTNUC_FK are called to set up the potentials used in $\Sigma^{\gtrless}$ and $\Sigma^{\mathrm{HF}}$.

Table 5
Variables of subroutine INIT_NUCL

| Program variables | Mathematical symbols | Comments |
|---|---|---|
| h2m2 | $\hbar^2/2m$ | $= 20.7355$ MeV fm$^2$ |
| anu | $\nu$ | degeneracy factor $= 4$ |
| hbarc | $\hbar c$ | $= 197.327$ MeV-fm |
| p0 | $p_0$ | a parameter, see Eq. (13) |
| pf | $p_f$ | a parameter, see Eq. (13) |

### 3.3.5. Subroutine EPPROP

This subroutine calculates the single particle energy propagator for each momentum grid point using the following:

$$U(\boldsymbol{p}) = \exp\left(\frac{i\epsilon(\boldsymbol{p})\Delta t}{\hbar}\right) \quad \text{if } p \leqslant idim,$$
$$= 1 \quad \text{if } p > idim,$$

where $\epsilon(\boldsymbol{p}) = \hbar^2 p^2/2m$. The input parameters of this subroutine are $ix$, $jx$, and, $kx$, which are discussed in the subroutine INIT_GRID and the momentum grid size *dpz*. The output are two arrays: *utk* which contains $U(\boldsymbol{p})$ and *utkc* which contains $(U(\boldsymbol{p}) - 1)/\epsilon(\boldsymbol{p})$. These arrays are used in the subroutine TSTEPI.

### 3.3.6. Subroutine TSTEPI

This subroutine propagates the Green functions $G^{\gtrless}(\boldsymbol{p}, t_1, t_2)$ by one time step along both time axes according to Eq. (1) assuming the current Green functions and the collision integrals $I^{\gtrless}_{1,2}(\boldsymbol{p}, t_1, t_2)$ are given. Because of the symmetries Eq. (7), we need to calculate $G^{\gtrless}$ only for either the upper triangle (region II) or the lower triangle (region I) of Fig. 1. We chose to calculate $G^<$ in the upper triangle plus the diagonal and $G^>$ in the lower triangle. The values in the other parts of the $t_1, t_2$ plane are obtained via the reflection symmetry expressed in Eq. (7). The Green functions are stored in a three-dimensional array as follows:

$$gre(iq, it1, it2) = G^<, \quad it1 \leqslant it2,$$
$$= G^>, \quad it1 > it2.$$

The time-stepping procedure is illustrated in Fig. 1. $T$ is the current time, i.e., the Green functions are supposed to have been calculated inside the square $[t_0 \leqslant t_1 \leqslant T, t_0 \leqslant t_2 \leqslant T]$. For each $t_0 \leqslant t_1 \leqslant T$, $G^<$ is time-stepped by integrating the second equation of Eq. (1) from $t_2 = T$ to $t_2 = T + \Delta t$, as indicated by the arrow in Fig. 1. As seen from Eq. (2) this requires $\Sigma^{\gtrless}(\boldsymbol{p}, t_1, t_2)$ to be known for $t_2 = T$ and $t_0 \leqslant t_1 \leqslant T$, i.e., along the line (1) of Fig. 1. Likewise one needs $G^{\gtrless}(\boldsymbol{p}, t_1, t_2)$ for $t_0 \leqslant t_2 \leqslant T$ and each $t_1 = t'$, i.e., along line (2). $G^>$ is time-stepped similarly from $t_1 = T$ to $t_1 = T + \delta t$ for each $t_0 \leqslant t_2 \leqslant T$.

The collision integrals $I_2^<(t_1, T)$ and $I_1^>(T, t_2)$ are calculated in the subroutine COLLIS and passed through a common block. At each time point on the square boundary in Fig. 1 but not on the diagonal, and for each momentum point, the following expressions are used in the time-stepping (the momentum label is suppressed):

$$G^<(t_1, T + \Delta t) = \exp\left(\frac{i\epsilon(\boldsymbol{p})\Delta t}{\hbar}\right) G^<(t_1, T) - \tfrac{1}{2}\left[I_2^<(t_1, T) + I_2^<(t_1, T + \Delta t)\right]$$
$$\times \frac{1 - \exp(i\epsilon(\boldsymbol{p})\Delta t/\hbar)}{\epsilon} + O(\Delta t^3), \quad t_0 \leqslant t_1 \leqslant T, \tag{16}$$

Table 6
Variables of subroutine COLLIS

| Program variables | Mathematical symbols | Comments |
|---|---|---|
| dgla | $I_1^>(\boldsymbol{p}, T, t_2)$ | the right-hand side of Eq. (1) at a fixed current time $T$ stored in a two-dimensional array |
| dgle | $I_2^<(\boldsymbol{p}, t_1, T)$ | the above comment is also applicable here |
| gla | $G^>(\boldsymbol{p}, t_1, T)$ or $G^>(\boldsymbol{p}, T, t_2)$ | at a fixed current time $T$, the values of $G^>$ are written into a local two-dimensional array |
| gle | $G^<(\boldsymbol{p}, T, t_2)$ or $G^<(\boldsymbol{p}, t_1, T)$ | the above comment is also applicable here |
| it | $t_1$ | see Fig. 1 |
| itp | $t_2$ | see Fig. 1 |
| sgla | $\Sigma_1^>(\boldsymbol{p}, T, t_2)$ | the correlation self-energy at a fixed current time $T$ stored in a two-dimensional array |
| sgle | $\Sigma_2^<(\boldsymbol{p}, t_1, T)$ | the above comment is also applicable here |
| smf | $\Sigma^{\mathrm{HF}}(\boldsymbol{p}, T)$ | the Fock self-energy at a fixed current time $T$ stored in a one-dimensional array |

$$G^>(T+\Delta t, t_2) = \exp\left(\frac{i\epsilon(\boldsymbol{p})\Delta t}{\hbar}\right) G^>(T, t_2) - \tfrac{1}{2}\left[I_1^>(T, t_2) + I_1^>(T+\Delta t, t_2)\right] \times \frac{1-\exp(i\epsilon(\boldsymbol{p})\Delta t/\hbar)}{\epsilon} + O(\Delta t^3), \quad t_0 \leqslant t_2 \leqslant T, \tag{17}$$

where $\epsilon = \hbar^2 p^2/2m$. For the time-point on the diagonal, $t_1 = t_2 = T$, the following equation is used:

$$G^<(T+\Delta t, T+\Delta t) = G^<(T,T) - \frac{i\Delta t}{2\hbar}\left[I_2^<(T+\Delta t, T+\Delta t) - I_1^>(T+\Delta t, T+\Delta t) + I_2^<(T,T) - I_1^>(T,T)\right] + O(\Delta t^3). \tag{18}$$

The reader interested in the derivation of the above equations may consult Appendix C. Eqs. (16)–(18) are solved by a two-pass iteration procedure. In the first pass, all the $T + \Delta t$ on the right-hand sides of the equations are replaced by $T$. This gives a first approximation for $G^<(\boldsymbol{p}, t_1, T+\Delta t)$ and $G^>(\boldsymbol{p}, T+\Delta t, t_2)$ which are then used to calculate $I_2^<(\boldsymbol{p}, t_1, T+\Delta t)$ and $I_1^>(\boldsymbol{p}, T+\Delta t, t_2)$. The final values of the propagated Green functions are then obtained from Eqs. (16)–(18). This iteration procedure is found to be sufficiently accurate to conserve energy and density. To improve the accuracy, it is most convenient to decrease the time step $\Delta t$. The program has the option of using a one-pass ($istf \neq 1$) or a two-pass ($istf = 1$) iteration.

### *3.3.7. Subroutine COLLIS*

This subroutine calculates the collision terms $I_2^<$ and $I_1^>$ in Eq. (2) with previously calculated $G^{\gtrless}$ from the subroutine TSTEPI, $\Sigma^{\gtrless}$ from the subroutine SIGMAS, and $\Sigma^{\mathrm{HF}}$ from the subroutine SIG_FK. The trapezoidal rule is used to perform the time integrals. As explained in the section on TSTEPI, the collision terms are calculated on the square boundaries defined by the current time $T$ in Fig. 1, i.e., for fixed $T$, $I_2^<(t, T)$ and then $I_1^>(T, t)$ are calculated for $t_0 \leqslant t \leqslant T$. Table 6 contains the variables defined in this subroutine.

### *3.3.8. Subroutine SIG_FK*

This subroutine calculates the Fock (exchange) part of the mean-field energy of a particle as given by Eq. (3). For spatially homogeneous systems, the first (Hartree) term in Eq. (3) is a constant independent of time and the state of the particle. We thus omit this term, the omission amounting to a redefinition of the origin of the energy axis. The second (Fock) term is a convolution of the mean-field potential and the instantaneous distribution $-iG^<(\boldsymbol{p}, t, t)$. This convolution integral is computed by Fast Fourier Transform. The (discrete) Fourier transform of the mean-field potential has been calculated in Subroutine POTNUC_FK or POTCOUL_FK and stored in the array *pot_fk*.

Table 7
Parameters of subroutine SIGMAS

| Program variables | Comments |
|---|---|
| nvec | see Table 3 |
| mid | = 1 + nvec/2 |
| cgla | temporary storage arrays for $G^>$ or its Fourier Transform. |
| cgle | temporary storage arrays for $G^<$ or its Fourier Transform |
| h | temporary storage arrays, see step 3 in Subsection 3.3.9 |
| h1 | temporary storage arrays, see step 3 in Subsection 3.3.9 |
| potl | the potential calculated by either POTCOUL or POTPAW shifted by mid |
| sgla | see Table 6 |
| sgle | see Table 6 |

In the present subroutine, the distribution is Fourier transformed, and the transform is then multiplied with that of the mean-field potential. The product is then inverse tranformed to give the mean-field energy $\Sigma^{\mathrm{HF}}(\boldsymbol{p}, t)$. The variables in this subroutine are also defined in Table 6.

In this and the next subroutines, the convolution/correlation integrals are computed with Discrete Fourier Transforms (DCT). When using this technique, the user should ensure that the '*end* effects' (the assumption of DCT that the input functions are periodic) and the '*aliasing* effects' (distortion of the Fourier spectrum due to under-sampling of the input function) have been reduced to an inconsequential level [17]. *Aliasing* is eliminated by using a sufficiently fine grid. The simplest treatment of the *end* effects is to pad the ends of the input functions with zeros. We adopt a practical approach here: we check the convergence of the program output, i.e., the distributions as functions of time, with respect to the grid size and the size of the zero pads. This issue is further discussed in Section 6 with the sample run results.

*3.3.9. Subroutine SIGMAS*

This subroutine calculates the self-energy using the expression given by Eq. (4). Being a double correlation/convolution integral, this expression is evaluated on a Cartesian grid using Fourier Transforms. Appendix A collects some relations regarding the correlation theorem. Table 7 gives a summary of the variables defined in this subroutine.

This subroutine uses $G^>(\boldsymbol{p}, T, \bar{t})$ and $G^<(\boldsymbol{p}, \bar{t}, T)$ previously calculated for $t_0 \leqslant \bar{t} \leqslant T$. The procedure for the evaluation of the integral given by Eq. (4) for $\Sigma^>(\boldsymbol{p}, T, \bar{t})$ and $\Sigma^<(\boldsymbol{p}, \bar{t}, T)$ is outlined in the following (in the rest of this subsection, the time dependence is suppressed).

The integral over $\boldsymbol{p}_2$ is calculated first. It is a correlation of two functions which is calculated by using the correlation theorem of Appendix 1. This involves the following steps:

(1) Calculate $\tilde{G}^<(\tilde{\boldsymbol{p}}) =$ the Fourier Transform of $G^<(\boldsymbol{p})$.
(2) Calculate $\tilde{G}^>(\tilde{\boldsymbol{p}}) =$ the Fourier Transform of $G^>(\boldsymbol{p})$.
(3) Calculate $h_1(-\bar{\boldsymbol{p}}) =$ the Fourier inverse of $[\tilde{G}^<(\tilde{\boldsymbol{p}})\tilde{G}^>(-\tilde{\boldsymbol{p}})]$, where $\bar{\boldsymbol{p}} = \boldsymbol{p} - \boldsymbol{p}_1$.
(4) Shift the argument of $h_1^<(\bar{\boldsymbol{p}})$ by half a period, as required by the definition of the Fast Fourier Transform (Appendix B).

Next the integral over $\boldsymbol{p}_1$ is done. This is also broken into the following steps:

(1) Calculate $h(\bar{\boldsymbol{p}}) = V(\bar{\boldsymbol{p}})^2 h_1(\bar{\boldsymbol{p}})$. The potential is symmetric under reflection about the origin: $V(-\boldsymbol{p}) = V(\boldsymbol{p})$ for any $\boldsymbol{p}$.
(2) Calculate $\tilde{h}(\tilde{\bar{\boldsymbol{p}}}) =$ the Fourier Transform of $h(\bar{\boldsymbol{p}})$.

(3) Calculate the Fourier inverse of $[\tilde{h}(-\tilde{\boldsymbol{p}})\tilde{G}^{>}(\tilde{\boldsymbol{p}})]$, which, after a half-period shift (see Step 4 above) and a multiplication by the degeneracy factor $\nu$, gives $\Sigma^{>}(\boldsymbol{p}, T, \bar{t})$.
(4) Calculate the Fourier inverse of $[\tilde{h}(\tilde{\boldsymbol{p}})\tilde{G}^{<}(\tilde{\boldsymbol{p}})]$, which, after a half-period shift and a multiplication by the degeneracy factor $\nu$, gives $\Sigma^{<}(\boldsymbol{p}, \bar{t}, T)$.

It is important to note that, according to Eqs. (1) and (2), only the values of $\overset{>}{\Sigma^{<}}$ on the square boundary in Fig. 1 are needed for the propagation of $\overset{>}{G^{<}}$ at any time $T$. In other words, one of the two time variables in $\overset{>}{\Sigma^{<}}$, which is needed for the time-stepping, is the current time $T$. So $\overset{>}{\Sigma^{<}}$ are stored as two two-dimensional (one momentum and one time) arrays and are updated at each time step.

Depending on the computing environment, either the Fast Fourier Transform routine CFFT3D in the SGI library or CCFFT3D in the CRAY library is called. See Section 5 for more details.

*3.3.10. Subroutine POTPAW*

This subroutine, called from INIT_NUCL calculates the square of the two-nucleon potential to be used for calculating $\overset{>}{\Sigma^{<}}$ in SIGMAS. The potential is a Gaussian [10] function of the momentum transfer $\boldsymbol{p}$,

$$V(\boldsymbol{p}) = \pi^{3/2}\eta^3 V_0 \exp(-\tfrac{1}{4}\eta^2 p^2), \tag{19}$$

where $V_0$ and $\eta$ are the strength and range scale constants, respectively. The values chosen from these constants are [10] $V_0 = 453$ MeV and $\eta = 0.57$ fm.

The input of this subroutine are *idim*, *dpz*. The output is the array $pot(i, j, k)$, the potential squared at each Cartesian point within the sphere of radius *idim*.

*3.3.11. Subroutine POTNUC_FK*

This subroutine, called from INIT_NUCL, calculates the effective potential to be used for the calculation of the Fock mean-field energy in SIG_FK in the nuclear case. We choose the potential given by [18] (we denote it by $V_{\mathrm{fk}}$ to indicate it is different from the potential used in SIGMAS)

$$V_{\mathrm{fk}}(\boldsymbol{p}) = \frac{2C}{\rho_0}\frac{1}{1 + p^2/\lambda}, \tag{20}$$

where $\rho_0$ is the empirical saturation density of zero-temperature nuclear matter ($= 0.16$ fm$^{-3}$), and $C$ and $\lambda$ are the strength and range scale constants, respectively. The values chosen for these constants are [18] $C = 64.95\nu$ MeV and $\lambda = 4.4378$ fm$^{-2}$.

This potential is evaluated for all $\boldsymbol{p}$ in the box and Fourier transformed. The transform elements are stored in the array *pot_fk*.

*3.3.12. Subroutine POTCOUL*

This subroutine, called from INIT_ELEC, calculates the square of the statically screened Coulomb potential,

$$V(\boldsymbol{p}) = \frac{4\pi e^2}{\epsilon_b(p^2 + \kappa^2)}, \tag{21}$$

where $e$ is the electronic charge, $\epsilon$ is the background dielectric constant, $\boldsymbol{p}$ is the momentum transfer, and $\kappa$ is the screening parameter which is either assigned a set value or calculated from the electron distribution at each time instant by the subroutine CALAKP.

*3.3.13. Subroutine POTCOUL_FK*

This subroutine, called from INIT_ELEC, calculates the potential to be used for the calculation of the Fock mean-field energy in SIG_FK in the electronic case. The potential used here is the unscreened Coulomb potential,

$$V_{\mathrm{fk}}(\boldsymbol{p}) = \frac{4\pi e^2}{\epsilon_b p^2}. \tag{22}$$

The background dielectric constant $\epsilon_b$ is the same as that in the previous subsection. At the origin, where the Coulomb potential is singular but integrable, we replace this expression by the average of the potential over one elementary grid cell,

$$V_{\rm fk}(\boldsymbol{p}=0)=\frac{8\pi e^2(\pi+0.69546)}{\epsilon_b \Delta z^2}, \tag{23}$$

where $\Delta z$ is the distance between two grid points along one coordinate axis.

This potential is evaluated for all $\boldsymbol{p}$ in the box and Fourier transformed. The transform elements are stored in the array *pot_fk*.

### 3.3.14. Subroutine CALAKP

This subroutine is called only if the user sets the parameter *isck* to 1. It calculates the static screening parameter $\kappa$ used in the subroutine POTCOUL from the instantaneous electron distribution,

$$\kappa^2=\zeta\int_0^\infty \frac{f(p_x,p_y,p_z)}{p^2}\,dp_x\,dp_y\,dp_z, \tag{24}$$

where $\zeta=(2\nu/\pi)(e^2/\epsilon_b)(m/\hbar^2)$. This expression is derived by taking the long-wavelength limit of the static Lindhard expression for the dielectric function, and assuming a spherically symmetric distribution. This expression is of course only a rough estimate of how the effects of screening change with the evolving distribution. It should also be noted that since it effectively leads to a time-dependent potential, the total energy of the system may not be conserved. A proper treatment of time-dependent screening would seem to need to at least include a time-dependent Random-Phase-Approximation for the self-energy [6].

If *isck* is set to 0, a preset value for $\kappa$ is used throughout the calculation.

### 3.3.15. Subroutine OUT_STAT

This subroutine calculates the particle density distribution, particle density, the kinetic energy, the potential energy, the quadrupole moment, and, for the electron system, the plasma frequency.

The particle density distribution, $f(\boldsymbol{p},t)$, is essentially the time-diagonal entry of $G^<$ as shown by Eq. (5).

The density is computed using the following expression:

$$n(t)=\frac{\nu}{(2\pi)^3}\int d^3p\, f(\boldsymbol{p},t). \tag{25}$$

The kinetic energy density is computed using the following expression:

$$\big\langle T(t)\big\rangle=\frac{\nu}{(2\pi)^3}\int d^3p\, f(\boldsymbol{p},t)\frac{\hbar^2p^2}{2m}. \tag{26}$$

The correlation potential energy density is computed using the following expression:

$$\big\langle V(t)\big\rangle=-\frac{i}{2}\frac{\nu}{(2\pi)^3}\int d^3p\, I_1^<(\boldsymbol{p},t,t). \tag{27}$$

The mean field (Fock) potential energy density is computed using the following expression:

$$\big\langle V_{\rm fk}(t)\big\rangle=\frac{1}{2}\frac{\nu}{(2\pi)^3}\int d^3p\, f(\boldsymbol{p},t)\Sigma^{\rm HF}(\boldsymbol{p},t). \tag{28}$$

The quadrupole moment of the distribution is computed using the following expression:

$$Q(t)=\frac{1}{n}\sqrt{\frac{5}{16\pi}}\frac{\nu}{(2\pi)^3}\int d^3p f(\boldsymbol{p},t)\big[2p_z^2-(p_x^2+p_y^2)\big]. \tag{29}$$

Table 8
Variables of subroutine OUT_STAT

| Program variables | Mathematical symbols | Comments |
|---|---|---|
| densy or oden | $n$ | the density given by Eq. (25) |
| fke or oke | $\langle T\rangle$ | the kinetic energy density given by Eq. (26) |
| pe_corr or ope_corr | $\langle V\rangle$ | the correlation potential energy density given by Eq. (27) |
| pe_mf or ope_mf | $\langle V_{\rm fk}\rangle$ | the mean field energy density given by Eq. (28) |
| ote | $\langle T\rangle + \langle V\rangle + \langle V_{\rm fk}\rangle$ | the total energy density of the system |
| qua or oqua | $Q$ | the quadrupole moment given by Eq. (29) |
| omega_pl | $\omega_{pl}$ | the plasma frequency given by Eq. (30) |

The plasma frequency is calculated using the following expression:

$$\omega_{pl} = \sqrt{\frac{4\pi e^2 n}{\epsilon_b m}}. \tag{30}$$

We note that this program is not set up to study the effects of density oscillations [6]. As a physical time scale of the system, the plasma frequency is recorded here only for reference.

An option exists for outputting snapshots of momentum distributions during its relaxation. When the input parameter *nplot* is set to be greater than zero, it gives the number of time instants when the distribution is desired. The input array $(iplot(i), i = 1, nplot)$ gives these time instants in units of time steps. At each such time point, the grid momenta along any axis and the distributions along the $x$ and $z$ axes are written into a file in a three-column format. A different output channel, running from 51 to 60 (which can of course be changed as desired) on the local operating system, is opened for each time point.

Table 8 shows the program's variable names assigned to these quantities.

*3.3.16. Subroutine PWRITE*

This subroutine writes the input parameters and the output of the program.

## 4. Verifications

As far as the authors know, there are no exact solution, analytic or numerical, to the general quantum nonequilibrium evolution problem for any realistic models of the electron plasma or nuclear matter (in three dimensions). Yet certain tests on our program, and the approximation it is based on, can be carried out in some limiting cases by comparing our results against those from other approximation schemes that are considered to be appropriate in those limits. Such tests, including the following, have been performed by us and our colleagues:

(1) In the weak-correlation limit, our program's results agree with the solutions to the corresponding Boltzmann equation [11].
(2) In the weak-correlation and weak-degeneracy limit, the results of a program using a similar algorithm as ours agree with those from molecular dynamics calculations [19].

For the general case away from these limits, we monitor the conservation of the particle density and the total energy. As explained in Section 1, the approximation that we have chosen for the self-energy belongs to the class of

[6] Two possible extensions of the program to study density fluctuations are (1) include an external field that induces the fluctuations and/or (2) adopt the Random-Phase-Approximation for $\overset{>}{\Sigma}{}^{<}$.

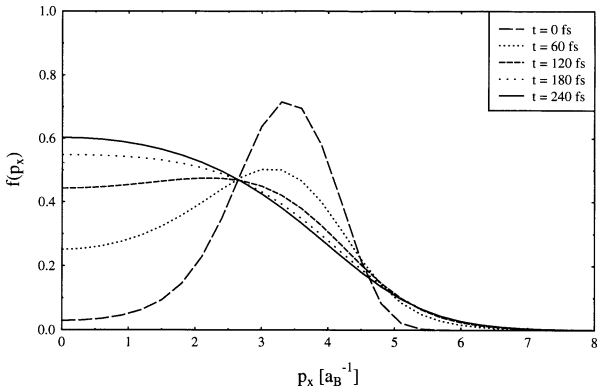


Fig. 2. The momentum distribution along the $x$-axis in the semiconductor run at various time instants.

'conserving' approximations, i.e., the Kadanoff–Baym equation conserves the system's total (kinetic + potential) energy and density with this self-energy [2,10]. The degree to which these quantities are numerically conserved, as shown in the displayed results in Section 6, serves as an indicator of the numerical accuracy of the calculations. The user will find that, when *istf* is set to 0 (one-pass time-stepping), although the density is still well conserved, the total energy conservation deteriorates.

It would be interesting to compare our nonequilibrium program in the limit of equilibrium with some Monte Carlo or other ground-state result. But as our interactions are to be regarded as effective interactions any comparison of this kind would be meaningless.

## 5. Program on different machines

The program has been run on Cray and SGI computers. Each of these computers may use a different library to perform the FFT task as we have discussed in Section 3.3.I. The FFT subroutines are initialized first at the beginning of the main section of the program. The call to the FFT routine CFFT3D is in subroutine SIGMAS. Subroutine CFFT3D either performs the FFT calculations for the SGI machine, or calls the FFT subroutine CCFFT3D for the Cray machine. The user must delete the last two subroutines of the program (CFFT3DI and CFFT3D) if the SGI computing environment is in use.

## 6. Test runs

In the Test Run Output, we display results from sample runs using the program. The input parameters are listed, and then the total density, quadrupole moment, the kinetic, correlation, mean field, and total energies as functions of time, as calculated according to the expressions Eqs. (25)–(29) in Section 3.3.15, are displayed. In the semiconductor case, we also display the momentum distributions along the positive $p_x$-axis (i.e., for $p_y = P_z = 0$) at several time instants (Fig. 2).

### *6.1. Semiconductor calculations*

The expression for the initial distribution is given by Eq. (12), the values of parameters being given in the Test Run Output. A Gaussian shell in momentum space, it models the momentum distribution of electrons immediately after excitation from the valence bands. The tabulated quadrupole moment and the plotted snapshots

of the distribution (Fig. 2) show the manner in which the distribution relaxes in the angular and radial directions respectively. The growth of the correlation energy (and the kinetic energy) is an indicator of the buildup of correlations among the electrons. For a more detailed discussion of the physical context within which this example was considered, the reader is referred to [7].

The convergence of the results with respect to grid size, and the aliasing effects in using the FFT for computing convolution/correlation integrals, have been examined with a slightly smaller box size: $dpz = 0.3a_B^{-1}$, $idim = 21$, $nvec = 48$. Halving the grid size and doubling the number of grid points – $dpz = 0.15a_B^{-1}$, $idim = 42$, $nvec = 96$ – results in a $< 1\%$ change in the distribution. The 'end effects' in using the FFT have been examined in another set of runs with the same input as the following displayed run except for one parameter: $\gamma_2$ (rgam2) is set to zero. This makes the initial distribution spherically symmetrical, allowing us to use the less time- and memory-demanding spherically symmetrical storage mode. A variation of array dimension *nvec* from 60 to 80 has been found to lead to a $< 1\%$ change in the distribution throughout the run. The numerical accuracy is not expected to vary significantly between the spherically symmetric case and the following cylindrically symmetric case.

### *6.2. Nuclear calculations*

The initial distribution (13) with temperature equal to zero is used for this sample run. It describes two counterstreaming slabs of nuclear matter in the center-of-mass frame. In the context of heavy ion collisions, in which a projectile nucleus hits a stationary target, the chosen value of the momentum boost $p_0$ in this run corresponds to a laboratory-frame energy of 400 MeV per nucleon in the projectile. The energies displayed in the Test Run Output are the energies per nucleon instead of the energies per unit volume. For a more detailed discussion of the physics, the reader is referred to [10,11].

The convergence of the results with respect to grid size, and the aliasing effects in using the FFT for computing convolution/correlation integrals, have been examined. Halving the grid size and doubling the number of grid points – $dpz = 0.15a_B^{-1}$, $idim = 42$, $nvec = 96$ – results in a $< 1\%$ change in the distribution. The 'end effects' in using the FFT have been examined. A variation of array dimension *nvec* from 48 to 60 has been found to lead to a $< 1\%$ change in the distribution.

## Acknowledgements

Some calculations and testing were made with the Cray computer at Pittsburgh's Supercomputer Center and with the SGI at the University of Arizona. This work was supported in part by National Science Foundation Grant No. PHY-9722050.

## Appendix A. Correlation relations

Let $g(y)$ and $f(y)$ be two functions and let $G$ and $F$ denote their Fourier Transforms, respectively. The correlation integral (in one dimension) is defined by

$$w(x) = \int_{-\infty}^{\infty} g(y) f(x+y)\, dy$$
$$= \int_{-\infty}^{\infty} g(y) \left[ \int_{-\infty}^{\infty} F(t) \exp(-i(x+y)t) \frac{dt}{2\pi} \right] dy$$

$$= \int_{-\infty}^{\infty} F(t)G(-t)\exp(-ixt)\,\frac{dt}{2\pi}.$$

Changing $x$ to $-x$ and $t$ to $-t$, we obtain the following expression:

$$w(-x) = \int_{-\infty}^{\infty} F(-t)G(t)\exp(-itx)\frac{dt}{2\pi}.$$

We use (the three-dimensional generalization of) this expression in the evaluation of the self-energy $\Sigma$.

## Appendix B. Numerical FFT convention

The Discrete Fourier Transform (in one dimension) on the machines that we have used is defined by

$$F_n = \frac{1}{s}\sum_{m=0}^{N-1} f_m \exp\left(-\frac{2\pi mni}{N}\right), \quad n = 0,\ldots,N-1,$$

where $N$ is the length of the input array. For the forward FFT, $s = 1$. The same expression holds for the backward FFT with the following changes: $-i$ is replaced by $i$ in the argument of the exponential function, $F$ and $f$ are interchanged, and $s = N$. With this definition of the Discrete Fourier Transform, the Correlation Theorem differs from the usual form by some permutations of indices. A straightforward derivation yields the following version of the theorem.

If a correlation integral of the functions $F$ and $G$ is defined in the continuum by

$$w(p) = \frac{1}{2\pi}\int f(p')g(p'-p)\,dp',$$

then its discretized approximant, with Fourier Transforms defined above, reads

$$W_n = \left(\frac{\Delta p}{2\pi}\right)h_{n'}, \quad n = 0,\ldots,N-1,$$

where $n' = (n+N/2) \bmod (N)$ and $h$ is the inverse transform of the function $H$ which is the product of the forward transforms $F$ and $G$ of $f$ and $g$, respectively,

$$H_m = F_m G_{N-m}, \quad m = 0,\ldots,N-1.$$

## Appendix C. Derivation of Eqs. (16)–(18)

### *C.1. Derivation of Eqs. (16) and (17)*

Suppressing the momentum label, we may write the Green function as

$$G^<(t_1,t_2) = g^<(t_1,t_2)\exp\left(\frac{i\epsilon(\boldsymbol{p})t_2}{\hbar}\right).$$

Substituting this expression into the second equation of (1), we get

$$-i\hbar\frac{\partial}{\partial t_2}g^<(t_1,t_2) = \exp\left(\frac{i\epsilon(\boldsymbol{p})t_2}{\hbar}\right)I_2^<(t_1,t_2).$$

We integrate this equation over $t_2$ from $T$ to $T+\Delta t$ to get

$$g^<(t_1, T+\Delta t) - g^<(t_1, T) = \frac{i}{\hbar} \int_T^{T+\Delta t} dt' \exp\left(\frac{i\epsilon(\boldsymbol{p})t'}{\hbar}\right) I_2^<(t_1, t').$$

$I_2^<$ is then approximated by the average of its values at $T$ and $T+\Delta t$ and pulled out of the integral. Performing the remaining integral of the exponential function, and multiplying through by $\exp(i\epsilon(\boldsymbol{p})(T+\Delta t)/\hbar)$, we obtain Eq. (16). Eq. (17) is obtained in a similar way.

*C.2. Derivation of Eq. (18)*

Subtracting the second equation from the first in (1), the following equation is obtained:

$$i\hbar\left(\frac{\partial}{\partial t_1} + \frac{\partial}{\partial t_2}\right) G^<(t_1, t_2) = I_1^<(t_1, t_2) - I_2^<(t_1, t_2).$$

By introducing the new variables $T = (t_1 + t_2)/2$, $t = t_1 - t_2$, one can replace $\frac{\partial}{\partial t_1} + \frac{\partial}{\partial t_2}$ by $\frac{\partial}{\partial T}$. Then, by integrating this equation from $T$ to $T+\Delta t$ along the time diagonal ($t_1 = t_2$) using the trapezoidal rule, Eq. (18) is obtained.

## References

[1] J. Schwinger, J. Math. Phys. 2 (1961) 407.
[2] L.P. Kadanoff, G. Baym, Quantum Statistical Mechanics (Benjamin, New York, 1962).
[3] L.V. Keldysh, ZHETF 47 (1964) 1515 [Sov. Phys. JETP 20 (1965) 235].
[4] R.A. Craig, J. Math. Phys. 9 (1968) 605.
[5] D.F. DuBois, in: Lectures in Theoretical Physics, Vol. 9c, W.E. Brittin, ed. (Gordon and Breach, New York, 1967) p. 469;
B. Bezzerides, D.F. DuBois, Ann. Phys. 70 (1972) 10.
[6] H. Haug, A. Jauho, Quantum Kinetics in Transport and Optics of Semiconductors (Springer, Berlin, 1996).
[7] R. Binder, H.S. Köhler, M. Bonitz, N.H. Kwong, Phys. Rev. B 55 (1997) 5110.
[8] W. Schäfer, J. Opt. Soc. Am. B 13 (1996) 1291.
[9] N.H. Kwong, M. Bonitz, R. Binder, H.S. Köhler, Phys. Status Solidi B 206 (1998) 197.
[10] P. Danielewicz, Ann. Phys. 152 (1984) 239–304; 304–326.
[11] H.S. Köhler, Phys. Rev. C 51 (1995) 3232; Phys. Rev. E 53 (1996) 3145.
[12] W. Botermans, R. Malfliet, Phys. Rep. 198 (1990) 115.
[13] P. Bozek, Phys. Rev. C 56 (1997) 1452.
[14] E. Calzetta, B.L. Hu, Phys. Rev. D 37 (1988) 2878.
[15] S. Mrówczyński, P. Danielewicz, Nucl. Phys. B 342 (1990) 345;
S. Mrówczyński, U. Heinz, Ann. Phys. 229 (1994) 1;
M. Schönhofen, M. Cubero, B.L. Friman, W. Nörenberg, Gy. Wolf, Nucl. Phys. A 572 (1994) 112.
[16] H.S. Köhler, R. Binder, Contrib. Plas. Phys. 37 (1997) 167.
[17] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in Fortran (Cambridge Univ. Press, Cambridge, 1992).
[18] G.M. Welke, M.Prakash, T.T.S. Kuo, S. Das Gupta, Phys. Rev. C 38 (1988) 2101.
[19] M. Bonitz, D. Semkat, D. Kremp, Phys. Rev. E 56 (1997) 1246.

**TEST RUN OUTPUT**

*The input parameters of the program: Semiconductor calculations*

dpz = 0.300000E+00 dft = 0.400000E+01
ntimes = 60 time0 = 0.000000E+00
number momentum mesh points per dimension = 57
exciton first radius (aindb) = 0.132000E+03
Rydberg in eV (eryd) = 0.420000E−02
electron effective mass (amee) = 0.670000E−01
background dielectric constant (epsb) = 0.129980E+02
initial distribution parameters:
ra = 0.120000E+01 rhb = 0.500000E−01 rgam = 0.194000E−01
rgam1 = 0.685000E+01 rgam2 = 0.210000E+01
options selected:
cylindrical symmetry, mod_store = 3
two passes per time step, istf = 1
electrons calculations, isy = 1
mean field included, ifk = 1
self-consistent static screening const.= 0.400000E+00
the above constant is not calculated, isck = 0
plasma frequency in $fs^{-1}$ = 0.472728E−01
dimension statement parameters:
nvec = 60 idim = 28
nq = 10225 ntl = 61

*The output of the program: Semiconductor calculations*

| Time (fs) | K. Energy (Ry-$a_B{}^{-3}$) | P. Energy (MF) (Ry-$a_B{}^{-3}$) | P. Energy (Corr.) (Ry-$a_B{}^{-3}$) | T. Energy (Ry-$a_B{}^{-3}$) | Density ($a_B{}^{-3}$) | Q. Moment ($a_B{}^{-2}$) |
|---|---|---|---|---|---|---|
| 0.000 | 0.1363E+02 | −.1413E+01 | 0.0000E+00 | 0.1221E+02 | 0.1406E+01 | −.7124E+00 |
| 4.000 | 0.1373E+02 | −.1402E+01 | −.1100E+00 | 0.1222E+02 | 0.1406E+01 | −.7112E+00 |
| 8.000 | 0.1395E+02 | −.1379E+01 | −.3573E+00 | 0.1222E+02 | 0.1406E+01 | −.7073E+00 |
| 12.000 | 0.1418E+02 | −.1352E+01 | −.6031E+00 | 0.1222E+02 | 0.1406E+01 | −.7020E+00 |
| 16.000 | 0.1438E+02 | −.1327E+01 | −.8208E+00 | 0.1223E+02 | 0.1406E+01 | −.6957E+00 |
| 20.000 | 0.1456E+02 | −.1304E+01 | −.1020E+01 | 0.1224E+02 | 0.1406E+01 | −.6889E+00 |
| 32.000 | 0.1501E+02 | −.1250E+01 | −.1519E+01 | 0.1224E+02 | 0.1406E+01 | −.6672E+00 |
| 48.000 | 0.1546E+02 | −.1203E+01 | −.2011E+01 | 0.1225E+02 | 0.1406E+01 | −.6375E+00 |
| 64.000 | 0.1579E+02 | −.1174E+01 | −.2362E+01 | 0.1225E+02 | 0.1406E+01 | −.6078E+00 |
| 80.000 | 0.1602E+02 | −.1157E+01 | −.2610E+01 | 0.1225E+02 | 0.1406E+01 | −.5785E+00 |
| 100.000 | 0.1621E+02 | −.1145E+01 | −.2820E+01 | 0.1225E+02 | 0.1406E+01 | −.5432E+00 |
| 120.000 | 0.1633E+02 | −.1139E+01 | −.2951E+01 | 0.1224E+02 | 0.1406E+01 | −.5095E+00 |
| 160.000 | 0.1644E+02 | −.1136E+01 | −.3080E+01 | 0.1223E+02 | 0.1406E+01 | −.4474E+00 |
| 200.000 | 0.1647E+02 | −.1139E+01 | −.3121E+01 | 0.1221E+02 | 0.1406E+01 | −.3924E+00 |
| 236.000 | 0.1647E+02 | −.1142E+01 | −.3129E+01 | 0.1220E+02 | 0.1406E+01 | −.3485E+00 |

*The input parameters of the program: Nuclear calculations*

dpz = 0.300000E+00 dft = 0.200000E+00
ntimes = 60 time0 = 0.000000E+00
number momentum mesh points per dimension = 43
initial distribution parameters:
p0 = 0.220000E+01 pf = 0.140000E+01
options selected:
cylindrical symmetry, mod_store = 3
two passes per time step, istf = 1
nuclear calculations, isy = 2
mean field included, ifk = 1
dimension statement parameters:
nvec = 60 idim = 21
nq = 4620 ntl = 61

*The output of the program: Nuclear calculations*

| Time (fm/c) | K. Energy (MeV) | P. Energy (MF) (MeV) | P. Energy (Corr.) (MeV) | T. Energy (MeV) | Density ($fm^{-3}$) | Q. Moment ($fm^{-2}$) |
|---|---|---|---|---|---|---|
| 0.000 | 0.1233E+03 | −.6581E+02 | 0.0000E+00 | 0.5748E+02 | 0.3701E+00 | 0.3005E+01 |
| 0.200 | 0.1269E+03 | −.6474E+02 | −.4138E+01 | 0.5798E+02 | 0.3701E+00 | 0.2991E+01 |
| 0.400 | 0.1350E+03 | −.6222E+02 | −.1513E+02 | 0.5769E+02 | 0.3701E+00 | 0.2950E+01 |
| 0.600 | 0.1435E+03 | −.5942E+02 | −.2659E+02 | 0.5754E+02 | 0.3701E+00 | 0.2884E+01 |
| 0.800 | 0.1496E+03 | −.5711E+02 | −.3488E+02 | 0.5761E+02 | 0.3701E+00 | 0.2801E+01 |
| 1.000 | 0.1530E+03 | −.5546E+02 | −.3982E+02 | 0.5775E+02 | 0.3701E+00 | 0.2710E+01 |
| 1.200 | 0.1547E+03 | −.5430E+02 | −.4259E+02 | 0.5785E+02 | 0.3701E+00 | 0.2619E+01 |
| 1.400 | 0.1555E+03 | −.5348E+02 | −.4414E+02 | 0.5789E+02 | 0.3701E+00 | 0.2530E+01 |
| 1.800 | 0.1557E+03 | −.5239E+02 | −.4534E+02 | 0.5795E+02 | 0.3701E+00 | 0.2359E+01 |
| 2.400 | 0.1551E+03 | −.5136E+02 | −.4580E+02 | 0.5796E+02 | 0.3701E+00 | 0.2125E+01 |
| 3.000 | 0.1546E+03 | −.5062E+02 | −.4607E+02 | 0.5795E+02 | 0.3701E+00 | 0.1915E+01 |
| 4.800 | 0.1541E+03 | −.4928E+02 | −.4689E+02 | 0.5792E+02 | 0.3701E+00 | 0.1405E+01 |
| 7.200 | 0.1538E+03 | −.4865E+02 | −.4731E+02 | 0.5787E+02 | 0.3701E+00 | 0.9320E+00 |
| 9.600 | 0.1537E+03 | −.4852E+02 | −.4734E+02 | 0.5780E+02 | 0.3701E+00 | 0.6187E+00 |
| 11.800 | 0.1535E+03 | −.4852E+02 | −.4730E+02 | 0.5773E+02 | 0.3701E+00 | 0.4247E+00 |