UCRL ID 104093

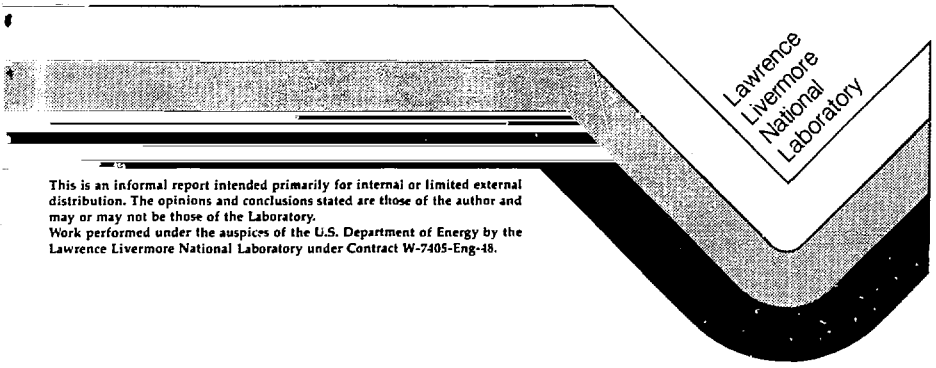
UCRL-ID--104093

DE90 011776

The DIPSI Computer Code User's Manual

Richard J. Procassini
Bruce I. Cohen

June 1, 1990



Price Code	Page Range
A01	Microfiche
Papercopy Prices	
A02	1- 10
A03	11- 50
A04	51- 75
A05	76-100
A06	101-125
A07	126-150
A08	151-175
A09	176-200
A10	201-225
A11	226-250
A12	251-275
A13	276-300
A14	301-325
A15	401-425
A16	351-375
A17	376-400
A18	401-425
A19	426-450
A20	451-475
A21	476-500
A22	501-525
A23	526-550
A24	551-575
A25	576-600
A99	601 & UP

The DIPSI Computer Code User's Manual

Richard J. Procassini*
Nuclear Engineering Department
University of California
Berkeley, CA 94720

Bruce I. Cohen
Magnetic Fusion Energy Program
Lawrence Livermore National Laboratory
Livermore, CA 94550

1 June 1990

Abstract

DIPSI (Direct Implicit Plasma Surface Interactions) is a one-dimensional, bounded particle-in-cell (PIC) simulation code designed to investigate the interaction of plasma with a solid surface, such as a limiter or divertor plate in a tokamak fusion device. Plasma confinement and transport may be studied in a system which includes an applied magnetic field (oriented normal to the solid surface) and/or a self-consistent electrostatic potential. The PIC code DIPSI is an offshoot of the PIC code TESS (Tandem Experiment Simulation Studies) which was developed to study plasma confinement in mirror devices. The codes DIPSI and TESS are direct descendants of the PIC code ES1 that was created by A. B. Langdon. This document provides the user with a brief description of the methods used in the code and a tutorial on the use of the code.

*Present Address: Lawrence Livermore National Laboratory, Livermore, CA 94550

1 Introduction

The particle simulation code DIPSI (Direct Implicit Plasma Surface Interactions) has been designed to model plasma confinement and transport in a phase space consisting of a single spatial dimension and two velocity components ($v_\parallel$ and $v_\perp$ relative to the spatial dimension). The code combines an implicit formulation of standard particle-in-cell (PIC) methods with Monte Carlo descriptions of Coulomb and charged/neutral collisional interactions. This report is intended as a guide to the use and modification of DIPSI. A brief description of the methods used in the code is provided in Section 2. Section 3 discusses the procedures for accessing, compiling and excuting the code. The important subroutines in the code are described in Section 4. A listing of the parameters and variables that may be input to the code is given in Section 5. Section 6 gives a brief description of the output diagnostics that are used in the code. Finally, Section 7 presents the results of two simulations that may be used to benchmark the code on new computer systems.

2 Methodology

This section provides a brief description of each of the major components of the DIPSI PIC code.

2.1 The Direct Implicit PIC Algorithm

The DIPSI PIC code follows the trajectories of individual charged particles in a single spatial dimension (for instance, along the open-magnetic field lines of a tokamak scrape-off layer $\hat{z}$) with components of the particle velocity that are parallel and perpendicular to the field lines ($v_\parallel$ and $v_\perp$). The guiding center of each particle is advanced according to the relativistic equations of motion

$$\frac{dz}{dt} = \frac{p_\parallel}{\gamma m} \tag{1}$$

$$\frac{dp_\parallel}{dt} = qE_\parallel - \frac{\mu}{\gamma}\nabla B_\parallel + \left(\frac{dp_\parallel}{dt}\right)_{coll} + \left(\frac{dp_\parallel}{dt}\right)_{cx} + \left(\frac{dp_\parallel}{dt}\right)_{ion} \tag{2}$$

$$\frac{d\mu}{dt} = \left(\frac{d\mu}{dt}\right)_{coll} + \left(\frac{d\mu}{dt}\right)_{cx} + \left(\frac{d\mu}{dt}\right)_{ion} \tag{3}$$

where x is the position along the field line, t is the time, $p_{\|} \equiv \gamma m v_{\|}$ is the axial momentum, $\gamma \equiv (1 - v^2/c^2)^{-1/2}$ is the relativistic factor, $\mu \equiv p_{\perp}^2/2mB_0$ is the magnetic moment, m and q are the particle rest mass and charge, and $E_{\|}$ and $B_{\|}$ are the electric and magnetic fields. Equations (2) and (3) indicate that the parallel momentum $p_{\|}$ and magnetic moment μ are modified as a result of Coulomb collisions between charged particles (*coll*), charge-exchange events between ions and neutrals (*cx*), and impact ionization of neutrals by electrons (*ion*).

A variable-damping, direct-implicit time-integration scheme [1] is used to solve the set of ordinary differential equations (1) to (3). A well-known derivative of this general scheme is the so-called D_1 scheme [2,3] which (neglecting the collisional, charge exchange and ionization effects) has the form

$$z^{n+1} = z^n + \frac{p_z^{n+1/2}\Delta t}{\gamma^{n+1/2}m} + \frac{qE^n(z^n)\Delta t^2}{2\left(\gamma^{n-1/2}\right)^3 m}\left[1 + \frac{2\mu B}{mc^2}\right] \tag{4}$$

$$p_z^{n+1/2} = p_z^{n-1/2} + \left[\bar{a}^n - \frac{\mu \nabla B}{\gamma^n}\right]\Delta t \tag{5}$$

where the recursive, electrostatic acceleration $\bar{a}_n$ is given by

$$\bar{a}^n = [\bar{a}^{n-1} + (q/m)\mathrm{E}^{n+1}(z^{n+1})]/2. \tag{6}$$

In this scheme the position advance is the same as that used in the explicit leap-frog method, while the implicit nature is to be found through use of the recursive acceleration $\bar{a}_n$ in the velocity advance.

The ambipolar, electrostatic potential is calculated via the direct-implicit form of Poisson's equation

$$-\nabla \cdot \left(1 + \sum_s \chi_s\right) \nabla \Phi(z) = 4\pi e\,(Zn_i(z) - n_e(z)) \tag{7}$$

where the summation is over species s (ions and electrons), n_s is the free-streaming or explicit density and χ_s is the implicit susceptibility, i. e. the implicit correction to the free-streaming charge density.

2.2 The Monte Carlo Coulomb Collision Algorithm

The binary particle collision algorithm that is incorporated into the DIPSI PIC code is fully relativistic, exclusive of radiation losses, and employs Monte Carlo techniques to simulate the scattering of plasma particles [4]. It is similar in operation to the nonrelativistic model developed

by Takizuka and Abe [5]. Since the code has only one spatial dimension (with three velocity components), it is not possible to simulate the details of Coulomb scattering based upon impact parameters. However, it is possible to scatter a particle's velocity through an angle which is chosen randomly from an appropriate distribution of angles. The mean scattering angle of this distribution is zero, and the width or variance of the distribution is a function of the particle charges, density, reduced mass and relative velocity, as well as the simulation time step (See Appendix B of Reference [6]. Particles within a given spatial grid cell are scattered off of each other without regard for the distance between the midpoints of these finite-size particles. The model is designed to handle any combination of projectile and target particles: electron-electron (e-e), electron-ion (e-i, which is equivalent to i-e) and ion-ion (i-i) scattering. This collision algorithm conserves the momentum and kinetic energy of the interacting particles. Reference [4] provides a detailed description of the Coulomb collision algorithm.

2.3 The Charged/Neutral Collision Algorithms

The charged plasma particles are also allowed to interact with a background neutral gas, which is treated as a continuum fluid. Both charge exchange and (electron) impact ionization events are included. Other charged/neutral interaction events may be included with little programming effort. The neutral particle density is specified to fall off exponentially with increasing distance from the solid surface boundary

$$n_n(z) = \hat{n}_n \exp\left(\frac{-[L-z]}{\mathcal{L}_n}\right) \tag{8}$$

where $\mathcal{L}_n$ is the neutral density e-folding length (which is equal to the average ionization mean-free path length in an actual plasma) and $\hat{n}_n$ is the peak neutral density adjacent to the solid surface. Both of these quantities are input parameters for this algorithm. This ad hoc representation of the neutral density assumes that the neutral particle density is maintained by the reflection of plasma flux from the solid surface. More general neutral density profiles are also possible with this algorithm. The peak neutral density is adjusted in time so as to maintain a constant ionization source rate, which is given by

$$S_{ion} = \int_{z'=z_{min}}^{L} n_e(z') n_n(z') \langle \sigma_{ion}(E_e) v_e \rangle dz' \tag{9}$$

where $n_e(z)$ is the electron density profile, σ_{ion} is the electron-imapct ionization cross section, E_e is the local electron kinetic energy and v_e is the local electron velocity.

The probability that a given electron will ionize a neutral atom, or that a given ion will undergo charge exchange with a neutral atom is given by [7,6]

$$P_{c,k} = 1 - \exp(-n_n(z)\sigma_c(E_{s,k})v_{s,k}\Delta t) \tag{10}$$

where c is the collision index (charge exchange or impact ionization), s is the species index (electrons or ions), k is the particle index, σ_c is the cross section for collision type c at kinetic energy $E_{s,k}$ and Δt is the simulation time step. The cross sections σ_{ion} and σ_{cx} that are used in (10) are simple fits to experimentally measured data. A given particle k undergoes a charged/neutral collision c if $R_N < P_{c,k}$, where R_N is a random number in the range (0,1). Once it is determined that an electron will ionize a neutral atom, energy balance relations are used to determine the kinetic energy of the scattered electron and new electron-ion pair. The energy of the new ion in a charge-exchange collision is chosen randomly. In either case, the kinetic energy of the neutral atom which undergoes the collision is chosen from the isotropic temperature kT_n, which is assumed to be spatially constant. The neutral gas temperature is the final input parameter that is required for the simulation of charged/neutral collisions. Chapter 4 of Reference [6] provides a more detailed discussion of the charge exchange and impact ionization models.

2.4 Plasma Injection Algorithms

For a distributed source, the velocity of each injected particle is chosen from the normalized source distribution functions $S_{\|,s}(z,v_\|)$ and $S_{\perp,s}(z,v_\perp)$ for each species s, which have the general forms

$$S_{\|,s}(z,v_\|)dv_\| = h(z)|v_\||^{\eta_s} \exp\left(\frac{-m_s v_\|^2}{2kT_{\|,s}}\right) dv_\| \tag{11}$$

and

$$S_{\perp,s}(z,v_\perp)dv_x dv_y = h(z) \exp\left(\frac{-m_s v_\perp^2}{2kT_{\perp,s}}\right) v_\perp dv_\perp d\theta \tag{12}$$

where $h(z)$ is the source intensity profile in the range $[0,1]$, k is Boltzmann's constant, and m_s, $T_{\|,s}$ and $T_{\perp,s}$ are the mass, parallel and perpendicular source temperatures for species s respectively. The exponent η_s allows one to choose either a Maxwellian ($\eta_s = 0$), flux ($\eta_s = 1$) or more general distribution function for the species s. The $v_\perp$ that follows the exponential in (12) is present because the perpendicular direction comprises two degrees of freedom (the $\hat{x}$ and $\hat{y}$ directions), such that $dv_x dv_y = v_\perp dv_\perp d\theta$, whereas $dv_z = dv_\|$. The perpendicular source

velocities are chosen randomly from the prescribed source function, while the parallel source velocities are chosen from a base-2 bit-reversed sequence for a "quiet" injection.

The source region is defined by the center location Z_{o_S} and the width ΔZ_S. The source intensity function that has been employed in DIPSI is $h(z) = 1$, although more general intensity profiles are possible. The particles are injected uniformly over the source region via

$$z_{inj} = Z_{o_S} + R_N \Delta Z_S \tag{13}$$

where R_N is a random number in the range $(-0.5, 0.5)$]. The electrons and ions are injected at the same location. Since the electron and ion velocities are, in general, not the same, it is possible to have charge separation that will result in an electrostatic source region potential variation.

The planar source algorithm is similar in many ways to the distributed source package. One major difference is the normalized source distribution function, which for particles emitted from a plane is a flux distribution

$$S_{\parallel,s}(v_\parallel)dv_\parallel = |v_\parallel| \exp\left(\frac{-m_s v_\parallel^2}{2kT_{\parallel,s}}\right) dv_\parallel \tag{14}$$

and

$$S_{\perp,s}(v_\perp)dv_x dv_y = \exp\left(\frac{-m_s v_\perp^2}{2kT_{\perp,s}}\right) v_\perp dv_\perp d\theta. \tag{15}$$

The positions of the injected particles are calculated in the following way. Each particle is injected at the left boundary ($z = 0$), but its position at the end of the injection routine is related to its velocity and when it was injected during the time step. The time step Δt is divided by the number of particles of the given species that are to be injected in that time step to give the time between injections as $\tau_{inj} = \Delta t / N_{inj}$. The position of the $i-th$ particle at the end of the injection routine is then given by

$$z_i = v_{\parallel,i} \left(N_{inj} - i\right) \tau_{inj}. \tag{16}$$

See Chapter 4 of Reference [6] for a more detailed discussion of the plasma injection algorithms.

2.5 The "Logical Sheath" Boundary Condition (DIPSI-LS Only!)

The "logical collector sheath" boundary condition [8] may be employed as the right-side system boundary (i. e. at the solid surface). This algorithm, which replaces the collector sheath by a

suitable boundary condition, reflects the proper number of electrons so as to maintain a zero net current at the surface of the electrostatically-floating solid surface. The collector sheath potential drop, which forms over $\sim 5\lambda_{De}$ adjacent to the solid surface, is replaced by a step jump in potential at $z = L$ (i. e. the collector sheath region is collapsed down to a plane). This equilibrium boundary condtion prevents the solid surface from charging up in time. All of the ions and the most energetic electrons that arrive at the right boundary are absorbed on the solid surface and removed from the system. Use of the logical sheath boundary condition alleviates the necessity of resolving fractions of an electron Debye length within the non-neutral collector sheath region. This allows one to obtain accurate simulations of bounded plasma systems with grid spacings that are larger than the electron Debye length ($\Delta z > \lambda_{De}$). This can translate into a significant reduction of the cost of the simulation for quasineutral bounded systems which are several orders of magnitude larger than λ_{De}.

3 Code Compilation and Execution

There are two versions of the DIPSI PIC code: a standard version (DIPSI) and one which employs the "logical sheath" boundary condition (DIPSI-LS). Both versions of the code run under the Basis System [9] on the Cray X/MP-2 supercomputer at the National Energy Resources Supercomputer Center (NERSC), formerly the National Magnetic Fusion Energy Computer Center (NMFECC).

3.1 Accessing The Files

All files associated with DIPSI (DIPSI-LS) are contained in the Cray LIB file DIPSILIB (DIPLSLIB). These files are stored in the NERSC CFS archival file storage system in directory /3054/DIPSIRJP. Within these LIB files, the FORTRAN source code is split up among several files:

(1) VAR.DIP - The Basis *variable description* file for package DIP; this file contains all of the global parameters (static quantities) and variables used in the code, separated into various *groups* (akin to COMMON blocks in ANSI FORTRAN)

(2) COLLSRC - The Monte Carlo, Coulomb COLLision algorithm routine

(3) ESPCSRC - The basic ElectroStatic Particle Code routines; this file also contains many of the diagnostic routines

(4) FRONTSRC - The Basis FRONT-end routines

(5) MISCSRC - MISCellaneous routines to check the validity of input data, process messages and check for runtime errors

(6) PLOTSRC - The diagnostic PLOTting routines

(7) SOUSRC - The plasma SOUrce injection, charge-exchange and ionization routines

Note - The files VAR.DIP, ESPCSRC and PLOTSRC in the LIB files DIPSILIB and DIPLSLIB are *NOT* the same. *To prevent confusion and frustration, it is suggested that only ONE set of files from these LIB files are in the working disk space at any given time.*

Each of the LIB files also contains several additional files, many of which are required to compile, load and execute the code. The root name displayed below by ⟨*root*⟩ is either DIPS for the files in DIPSILIB (standard version), or DIPL for the files in DIPLSLIB (logical sheath version):

- ⟨*root*⟩CFT - A COSMOS batch file that is used for compilation and loading
- ⟨*root*⟩EXE - The Cray X/MP-2 executable file
- ⟨*root*⟩XLIB - The user-defined library for the Cray X/MP-2
- ⟨*root*⟩OUT - A COSMOS batch file to rename out files after execution
- MAC.DIP - A file required by Basis during execution
- MPPL.IN - An input file required by Basis' MPPL preprocessor prior to compilation
- PACK.DIP - An input file required by Basis' CONFIG preprocessor prior to compilation

The other files contained in the LIB files are input, text output (ASCII) and graphic output (FR80) files, which will be described in detail in Section 7 below.

3.2 Compiling and Loading The Code

To create the DIPSI or DIPSI-LS executable file (i. e. compile the source code with the CFT compiler and load the resulting object file with LDR), simply enter the command:

COSMOS ⟨*root*⟩CFT / *t v*

where *t* and *v* are the *time* and *value* assigned for the given task on the Cray. This will produce the following files:

- ⟨*root*⟩EXE - The executable file
- ⟨*root*⟩LST - The CFT compilation listing
- ⟨*root*⟩MAP - The LDR load map

3.3 Executing The Code

To excute DIPSI or DIPSI-LS, simply enter the command:

⟨*root*⟩EXE READ DIPI⟨*runid*⟩ / *t v*

where DIPI⟨*runid*⟩ is the DIPsi Input file, with ⟨*runid*⟩ the 4 character run identification string consisting of three numbers and one letter (i. e. 101c). The code will be initialized and the input file read in, at which point the "DIPSI)" prompt will appear. To begin the run, enter the command:

RUN;END

During execution, the status of the run may be queried by typing:

STATUS

which will provide the user with the current time step number, the total time used to that point (CPU plus Input/Output plus SYStem time), and the number of electron and ion superparticles in the plasma. To interrupt the execution, and return control to the Basis user level (the "DIPSI)" prompt), type in:

STOP

To restart the code from the Basis user level (the "DIPSI)" prompt) and finish the run, type in the command:

STEP;FINISH;END

See the Basis user's manual [9] for more information on the commands STEP and FINISH. Upon successful completion of the run, the following files are found on disk ($\langle suffix \rangle$ is the single letter suffix or channel (A to E) on which the code was run):

- +DIPSEX$\langle suffix \rangle$ - The dropfile for the run
- F3$\langle runid \rangle$nn - The (family) of FR80 Graphics file(s) which constitutes the majority of the output for the run
- $\langle runid \rangle\langle suffix \rangle$ 1 - An ASCII (Hardcopy) output file containing a list of the input data and a complete timing of the run broken down by subroutine
- DIPSI%$\langle suffix \rangle$ - The logfile for the run

In most cases, it will be helpful to rename these files (except the FR80 file) to include the run identification string in the filename. To do this, simply enter the command (on the *SAME* suffix that the code was executed on):

COSMOS $\langle root \rangle$OUT WITH $\langle runid \rangle$ / *t v*

This procedure renames those files as follows:

- +DIPSEX$\langle suffix \rangle$ $\Longrightarrow$ +DIP$\langle runid \rangle$
- $\langle runid \rangle\langle suffix \rangle$1 $\Longrightarrow$ DIPH$\langle runid \rangle$
- DIPSI%$\langle suffix \rangle$ $\Longrightarrow$ LOG$\langle runid \rangle$

4 Important Subroutines

While the source code itself is very well documented (including a definition of the input and output arguments, and local variables in each subroutine), it is worthwhile to briefly describe the important subroutines in each of the source code files.

File COLLSRC

- COLLIDE - Coulomb scatters the charged particles off each other via a Monte Carlo, binary-particle algorithm

File ESPCSRC

- ACCUM - Performs the accumulation of the charge density $\rho(z)$ and susceptibility $\chi(z)$ to solve for the potential $\phi(z)$ at each time step
- EZFLUC - Determines the fluctuation spectrum in k-space of the axial electric field
- GETDF1D/GETDF2D - Computes one- and two-dimensional distribution functions from the discrete particles locations in velocity, space, etc.
- GETPHI - Calculates the electrostatic potential via *(i)* solution of the finite difference form of Equation (7) each time step, or *(ii)* as a (static) fit to a cubic spline at time step zero
- LOADV/LOADZ - Loads the particles at time step zero in velocity space (as a Maxwellian via use of subroutines MAXWEL1D/MAXWEL2D) and in configuration space
- MOMENT - Calculates various moments of the particle distribution functions each time step that time-history samples are taken (every NTSAMPLE time steps)
- PLOTDF/PLOTDIST - Sets up various distibution function and moment diagnostics for plotting
- POSTPUSH/PREPUSH - The particle trajectory time-integration, or *push*, subroutine that is separated into two half-advances

- *PUSH* - The "master" execution subroutine that advances the particle trajectories, collides the particles, etc.
- REPACK - Removes particles that have left the system from the particle arrays
- SAMPLE - Samples various quantities at regular intervals (every NTSAMPLE time steps) *during the run to generate the time history diagnostics*
- SETUP - The "master" initialization subroutine, which in turn calls SETB (defines the applied magnetic field profile), SETCON/SETPCON (defines the run parameters, general physical constants and species constants), SETCONV (defines the conversion constants between the physical input data and the internal code data) and SETGRID (defines the grid structure)

File FRONTSRC

The purpose of the functions and subroutines in this file are described in detail in the Basis system manual [9].

File MISCSRC

- ERRMES - Provides error message flow control
- MSGPROC - Provides message flow control to *(i)* check the status of the run (enter STATUS), or *(ii)* stop the run at the end of the current time step (enter STOP)
- VALIDATE - Checks the validity of the input data

File PLOTSRC

- FILMS - Generates color 16mm movies via the Dicomed camera when the variable IFILM = 1
- MOVIE - Plots the time history arrays at the end of the run

- MURAL - Plots miscellaneous data at the end of the run
- PORTRAIT - Plots the initial conditions at the beginning of the run
- REPORT - Prints the input data at the beginning of the run
- SNAPSHOT - Plots the diagnostics at regular intervals (every NTSNAP time steps) during the run

File SOUSRC

- CHARGX - Simulates charge-exchange events between ions and neutral gas particles
- IONIZE - Simulates the ionization of neutral gas particles due to impact by electrons
- NEUTSET - Normalizes the peak neutral particle density in order to maintain a constant electron-impact ionization rate
- PSOURCE - Injects thermal plasma from a planar source located at the symmetry plane ($z = 0$)
- STREAM - Injects thermal plasma from a planar source located at the solid surface ($z = L$)
- VSOURCE - Injects thermal plasma in a distributed (volumetric) source region

5 Input Parameters and Variables

The input data for the DIPSI PIC code consists of parameters (scalars or arrays which are set at the beginning of the run whose value remain constant, or static, throughout the run) and variables (scalars or arrays whose value can change during the course of the run). The input parameters are located at the beginning of the variable description file VAR.DIP, while the input variables are located within the various groups in VAR.DIP. The input parameters and variables for DIPSI are listed in Tables 1 and 2 respectively. Note that it is necessary to recompile the code in order to change any of the parameters listed in Table 1.

When running the code in the implicit limit ($\omega_{pe}\Delta t \gg 1$ and $\Delta z/\lambda_{de} \gg 1$), experience [10] has indicated that selecting the input parameters such that $v_{te}\Delta t/\Delta z \sim 0.3$, where v_{te} is the electron thermal velocity, will lead to good conservation of total-energy.

6 Code Output Diagnostics

The PIC code DIPSI (DIPSI-LS) produces both graphics and text output files. The text output is written to the ASCII file DIPH(*runid*), while the graphics (plotted) output is written to the FR80 file F3(*runid*)nn.

The output data in the FR80 output file(s) is separated into several sections, each of which is briefly described below:

- REPORT - This section of the output data begins with a "banner" page which lists the name of the code, the version number, the date and time the run began, the machine and suffix the run was performed on, and the run identification number and label. This is followed by a listing of the important run parameters grouped in terms of timing, geometry, magnetic field, general diagnostic/operational, Coulomb collision, charged/neutral collision, plasma injection and species particle data.
- PORTRAITS - After the input parameters for the run have been listed, the initial conditions are presented. The following diagnostics are plotted:
  - Profiles of the applied magnetic flux density $B(z)$ and gradient $dB(z)/dz$
  - Profiles of the electrostatic potential $\phi(z)$, charge density $\rho(z)$, electric field $E(z)$ and susceptibility $\chi(z)$
  - Distribution functions $f(v_{\|})$ and $f(v_{\perp})$, velocity space density contours and scatter plots $f(v_{\|}, v_{\perp})$ *for each species*
  - Velocity space density contours $f(v_{\|}, v_{\perp})$ and distribution functions $f(v_{\|})$ and $f(v_{\perp})$ within user-defined regional (spatial) windows *for each species*
  - Phase space scatter plots $f(z, v_{\|})$ and $f(z, v_{\perp})$ *for each species*
  - Profiles of the moments of the distribution functions *for each species*: density ($n(z)$ - zeroth), drift velocity ($\langle v_{\|} \rangle(z)$ - first), kinetic energy ($\varepsilon(z)$ - second), temperature

($T(z)$ - variance of second moment), kinetic energy flux ($q_{tot}(z)$ - third) and conduction/convection heat fluxes ($q_{cond}(z)$ and $q_{conv}(z)$ - derived from third moment)

- SNAPSHOTS - This "instantaneous" data is plotted at regular intervals during the run (every NTSNAP time steps). The diagnostics used for the snapshots are the same as those plotted in the PORTRAITS section, except that:
  - The magnetic flux density and gradient are not plotted
  - Time-averaged and lag-averaged potential profiles $\bar{\phi}(z)$, as well as an electric field fluctuation spectrum $\tilde{E}(k)$ are plotted after the instantaneous profiles of the potential, charge density, electric field and susceptibility
- MOVIES - The time-history diagnostics are plotted at the conclusion of the run. The histories of the following quanties are plotted:
  - The number of superparticles $N(t)$ *of each species*
  - The kinetic energy $\varepsilon(t)$ *of each species*
  - The electrostatic energy $E^2(t)/8\pi$ and total energy $\epsilon_{tot}(t) \equiv \sum_s \varepsilon_s(t) + E^2(t)/8\pi$
  - The deviation from exact energy conservation $\Delta\epsilon_{tot}(t)$
  - The charge on the surface of the absorbing solid surface $\sigma_w(t)$
  - The instantaneous and lag-averaged floating potential $\phi_w(t)$ and $\bar{\phi}_w(t)$
  - The peak neutral particle density $\hat{n}_n(t)$, as well as the charge-exchange and ionization rates $R_{cx}(t)$ and $R_{ion}(t)$
  - The total entropy $S_{tot}(t)$, as well as the entropy *of each species* $S_s(t)$
  - The mean-square velocity $\langle v^2 \rangle(t)$, mean axial velocity $\langle v_\parallel \rangle(t)$ and mean-square pitch angle $\langle \sin^2(\theta) \rangle(t) \equiv \langle (v_\perp/v)^2 \rangle(t)$ *of each species*
  - The number of superparticles *of each species* lost per time step $N_{lost}(t)$ *[DIPSI-LS Only!]*
  - The instantaneous and lag-averaged presheath potential drops $\phi_1(t)$ and $\bar{\phi}_1(t)$, and collector sheath potential drops $\Delta\phi_c(t)$ and $\overline{\Delta\phi}_c(t)$ *[DIPSI-LS Only!]*
- MURALS - Miscellaneous code diagnostics plotted at the end of the run. These include:

- Time-integrated profiles of volumetric plasma injection, charge exchange and ionization
- Maximum and time-averaged profiles of the superparticle density per grid cell

The data in the text file DIPH(*runid*) is broken up into a few sections. These are:

- REPORT - This section displays the same banner information and list of run parameters that is included in the FR80 graphics output file F3(*runid*)nn as discussed above.
- TIMING - This section provides a detailed breakdown of the cost of the run. The following data is given for each subroutine or function that is executed during the run:
  - The number of executions N_{exe}
  - The total CPU, Input/Output and System time required to execute the given subroutine or function N_{exe} times
  - The average CPU, Input/Output and System time required to execute the given subroutine or function (averaged over N_{exe} executions)
- *For DIPSI-LS Only*: The time step, as well as the number of electrons and ions lost that time step, $N_{lost,e}$ and $N_{lost,i}$, are printed out whenever $N_{lost,i} > N_{lost,e}$.

7 Test Cases

The input file (DIPI(*runid*)), ASCII (hardcopy) output file (DIPH(*runid*)), and graphics output file(s) (F3(*runid*)nn) for two example problems (differentiated by the run identification strings (*runid*)) are also included in the LIB files DIPSILIB and DIPLSLIB. These simulations are intended to illustrate DIPSI's rather extensive capabilities for modeling bounded-plasma systems. These simulations also serve as test cases on which to benchmark the code on other computer systems. Each of these test cases is briefly described below.

7.1 Test Case 1 - A Collisionless Plasma-Sheath Region

$\langle runid \rangle$ = 045a

(Standard Version: DIPSI)

The bounded-plasma system modeled in this example problem consists of a planar source located at the symmetry plane (the left boundary at $z = 0$) and an electrostatically-floating, fully-absorbing solid surface (the right boundary at $z = L$). A right-going half-Maxwellian of electrons and ions is injected at $z = 0$. The fully-ionized plasma system is assumed to be (Coulomb) collisionless. To obtain an equilibrium solution, it is necessary to rethermalize, or *reflux*, the electron velocities at the symmetry plane (at $z = 0$). See [11] for a detailed discussion of the use of the refluxing procedure in collisionless, bounded PIC simulations. This simulation is run *explicitly*: $\omega_{pe}\Delta t < 1$ (0.10) and $\Delta z/\lambda_{De} < 1$ (0.21). Therefore the electron plasma period and Debye length are resolved.

The results presented in files F3045Ann show that the electrostatic potential profile decreases monotonically as one approaches the solid surface. The total potential drop consists of a source sheath near $z = 0$ (since the thermal velocity of the injected electrons is larger than that of the injected ions), and a collector sheath drop near $z = L$ (because the solid surface maintains a net negative charge, even at steady state). The ions are shown to be accelerated towards the solid surface due to the potential structure. The majority of the electrons are reflected by the collector sheath drop, and subsequently refluxed when they reach the symmetry plane.

7.2 Test Case 2 - A Low-Recycling Tokamak Scrape-Off Layer

$\langle runid \rangle$ = 058b

(Logical Sheath Version: DIPSI-LS)

This example problem makes use of all of the physics packages that are incorporated into the DIPSI-LS PIC code. Warm plasma is injected into the scrape-off layer (SOL) in a distributed source region ($0 \leq z \leq L_S$) upstream in the vicinity the symmetry plane (at $z = 0$). Charged/neutral interactions (charge exchange and ionization) occur in a region adjacent to the bounding divertor plate (at $z = L$). The average ionization frequency is equal to 1% of the thermal (source) electron bounce frequency ($\langle \nu_{ion} \rangle = 0.01\nu_{be0}$, which corresponds to a low-level

of neutral particle recycling from the divertor plate. Coulomb collisions occur throughout the system (such that the refluxing procedure is *not* required) with the average (source) collision frequency equal to the thermal (source) electron bounce frequency $\langle \nu_c \rangle = \nu_{be0}$. Particle are specularly reflected at the symmetry plane: $-z \rightarrow z$ and $-v_\parallel \rightarrow v_\parallel$. This simulation is run *implicitly*: $\omega_{pe}\Delta t \gg 1$ (10.45) and $\Delta z/\lambda_{De} \gg 1$ (31.45). Since the electron plasma period and Debye length are not resolved, the logical sheath boundary condition is employed to obtain the correct electron dynamics near the plate (however, the collector sheath drop is *not* plotted on the potential profile diagnostics).

The results presented in file F3058B0X indicate that the potential structure is modified by the presence of charged/neutral collisions in the vicinity of the divertor plate. The quasineutral potential profile is no longer monotonically decreasing as one approaches the plate. The potential decreases in the warm-particle source region upstream, continues to decrease in the first-half of the collisional, source-free region. The potential then rises in the second-half of the source-free region and the majority of the charged/neutral interaction region, before falling off sharply near the entrance to the collector sheath. The charged/neutral collisions are also found to reduce the plasma flow velocity, as well as the kinetic energy and heat conduction fluxes to the divertor plate. *Simulations of this type are described in detail in Chapters 6 and 7 of* Reference [6].

Acknowldgements

The authors wish to acknowledge the efforts of M. E. Stewart (Lawrence Livermore National Laboratory) and J. C. Cummings (Princeton University), who assisted in programming the PIC codes TESS and DIPSI, and P. F. DuBois (Lawrence Livermore National Laboratory) for introducing us and the code to Basis. The development of these codes is also due in part to the efforts of R. P. Freis (Lawrence Livermore National Laboratory) and C. K. Birdsall (University of California at Berkeley). Funding for the devlopment of these codes in the form of grants was provided by the Institutional Research and Development Program, and the Plasma Physics Research Institute at the Lawrence Livermore National Laboratory, under the auspices of the U. S. Department of Energy, Contract Number W-7405-Eng-48. Additional funding was provided to the Plasma Theory and Simulation Group at the University of California at Berkeley by the U. S. Department of Energy under Contract Number DE-FG03-86ER53220.

Table 1

DIPSI Input Parameters				
Parameter	Units	Default Value	Data Type	Description
MAXG	cells	128	Integer	Maximum number of grid cells
MAXSP	specie^{-1}	6500	Integer	Maximum number of super-particles/specie
MAXTP	specie^{-1}	20	Integer	Maximum number of tracer particles/specie
MAXT	samples	750	Integer	Maximum number of time history samples
MAXMARK	points	10000	Integer	Maximum number of points plotted on a scatter plot
MAXNBSP	points	25	Integer	Maximum number of magnetic field spline data points
MAXNPSP	points	25	Integer	Maximum number of potential spline data points
MAXVSP	elements	2500	Integer	Maximum number of elements in volume-source arrays
MAXNSIG	elements	1000	Integer	Maximum number of elements in $\sigma_{cx}(E)$ and $\sigma_{ion}(E)$ arrays
NBIN1D	bins	129	Integer	Number of bins for 1-D velocity arrays
NBIN2D	bins	40	Integer	Number of bins for 2-D velocity arrays (NBIN2D x NBIN2D)

Table 2

DIPSI Input Variables

Variable	Units	Default Value	Data Type	Description
		Group STRINGS (Run Description Strings)		
RUNID		"000a"	Character*4	Identification label for the current run
RUNLABEL		"DIPSI Runlabel"	Character*30	Descriptive label for the current run
		Group RUNPARAM (Program Control Variables)		
IFILM		0	Integer	Film or plot mode indicator (0 = Plot , 1 = Film)
IRET		0	Integer	Switch for particle retherm-alization upon reflection (0 = No , 1 = Yes)
ITFIRST	step	1	Integer	Initial time step for the run
ITLAST	step	10000	Integer	Final time step for the run
NG	cells	MAXG	Integer	Number of grid cells
NT	steps	MAXT	Integer	Number of time steps for the run
NTFRAME	steps	10	Integer	Number of time steps between film frames
NTP	particles	MAXTP	Integer	Number of tracer particles
NTSNAP	steps	-1	Integer	Number of time steps between snapshots
NCOLLPP		1	Integer	Integer number of collisions per particle every NTCOLL time steps

Table 2 (continued)

Variable	Units	Default Value	Data Type	Description
NTCOLL	steps	1	Integer	Number of time steps between execution of Coloumb collision package
NTNEUT	steps	1	Integer	Number of time steps between neutral density normalization
NEXPAND	elements	500	Integer	Number of elements to expand particle arrays by during dynamic dimensioning
Group CONSTANT (Run Constants)				
DAMPFACT		1.0	Real	Implicit scheme adjustable-damping factor
DT	sec	1.0e-07	Real	Time step size
RMASS		0	Integer	Ion to electron mass ratio
ZNUM		1	Integer	Absolute value of ion to electron charge ratio
ZPSINT	cm	0.0	Real	Location of plasma-sheath interface
Group GRIDS (Grid Definition Variables)				
LENGTH	cm	800.0	Real	Total system length
LENSLAB	cm	0.0	Real	Length of initial plasma slab
Group BFIELD (Magnetic Field Variables)				
BLEFT	gauss	3000.0	Real	Magnetic field strength on left side of system
NBSP	points	0	Integer	Number of magnetic field spline data points

Table 2 (continued)

Variable	Units	Default Value	Data Type	Description
SPB(MAXNBSP)	gauss	MAXNBSP*0.0	Real	Magnetic field strength at spline data points
SPDBDZ(MAXNBSP)	gauss/cm	MAXNBSP*0.0	Real	First derivative of magnetic field ($dB(z)/dz$) at spline data points
SPZB(MAXNBSP)	cm	MAXNBSP*0.0	Real	Axial position of magnetic field spline data points
Group EFIELD (Potential and Electric Field Variables)				
FPHIS		0.5	Real	Lag-averaged, time-filter factor in the range [0,1]
NPSP	points	0	Integer	Number of potential spline data points
SPP(MAXNPSP)	V	MAXNPSP*0.0	Real	Potential at spline data points
SPDPDZ(MAXNPSP)	V/cm	MAXNPSP*0.0	Real	First derivative of potential ($d\phi(z)/dz$) at spline data points
SPZP(MAXNPSP)	cm	MAXNPSP*0.0	Real	Axial position of potential spline data points
Group ELECTRONS (Electron Species Variables)				
NE	cm^{-3}	0.0	Real	Initial electron particle density
NSPE		MAXSP	Integer	Initial/actual number of electron superparticles
NSPEQE		0	Integer	Predicted number of electron superparticles at equilibrium

Table 2 (continued)

Variable	Units	Default Value	Data Type	Description
TEMPPE	keV	1.0	Real	Initial/source perpendicular electron temperature
TEMPZE	keV	1.0	Real	Initial/source parallel electron temperature
COLLRATE		0.0	Real	Average electron Coulomb collision frequency times the time step
Group IONS (Ion Species Variables)				
NI	cm^{-3}	0.0	Real	Initial ion particle density
NSPI		MAXSP	Integer	Initial/actual number of ion superparticles
NSPEQI		0	Integer	Predicted number of ion superparticles at equilibrium
TEMPPI	keV	1.0	Real	Initial/source perpendicular ion temperature
TEMPZI	keV	1.0	Real	Initial/source parallel ion temperature
COLLRATI		0.0	Real	Average ion Coulomb collision frequency times the time step
Group HEATING (Charged/Neutral Interaction Variables)				
Charge Exchange Variables				
NSKIPCX	particles	1	Integer	Particle stride factor used in charge exchange process
SCXMAX	cm^2	0.0	Real	Maximum value of the charge exchange cross section: $\hat{\sigma}_{cx}$

Table 2 (continued)

Variable	Units	Default Value	Data Type	Description
Ionization Variables				
GAMBWION	keV	1.0e-2	Real	Breit-Wigner energy level width for the ionization cross section
KE1ION	keV	4.0e-2	Real	Minimum kinetic energy for the ionization cross-section plateau
KE2ION	keV	1.0e-1	Real	Maximum kinetic energy for the ionization cross-section plateau
KEMINION	keV	1.5e-2	Real	Threshold kinetic energy for ionization
NSKIPION	particles	1	Integer	Particle stride factor used in ionization process
SIONMAX	cm^2	0.0	Real	Maximum value of the ionization cross section: $\hat{\sigma}_{ion}$
Stream Gun Variables				
SGRATE	pairs/time step	0.0	Real	Stream gun particle-pair injection rate
SGENRGY	keV	0.0	Real	Stream gun average particle energy
SGTEMP	keV	0.0	Real	Stream gun particle thermal spread
Volume-Source Injection Variables				
VSRATE	pairs/time step	0.0	Real	Volume source particle-pair injection rate

Table 2 (continued)

Variable	Units	Default Value	Data Type	Description
VSVDE	cm/sec	0.0	Real	Volume source axial electron drift velocity
VSVDI	cm/sec	0.0	Real	Volume source axial ion drift velocity
ZCTRVS	cm	-1000.0	Real	Volume source midpoint location
ZWVS	cm	50.0	Real	Volume source half-width
Planar-Source Injection Variables				
PSRATE	pairs/time step	0.0	Real	Planar source particle-pair injection rate
PSVDE	cm/sec	0.0	Real	Planar source axial electron drift velocity
PSVDI	cm/sec	0.0	Real	Planar source axial ion drift velocity
Shared Variables				
NNMAX1	cm^{-3}	0.0	Real	Initial peak neutral particle density for 1st neutral profile
ZWCX1	cm	10.0	Real	Neutral density e-folding length for 1st neutral profile
CXTEMP1	keV	0.0	Real	Neutral particle temperature for 1st neutral profile
IVZSDF		1	Integer	Parallel velocity source distribution function index (1 for $S(v_{\|\|}) = f_{Max}(v_{\|\|})$; 2 for $S(v_{\|\|}) = v_{\|\|} f_{Max}(v_{\|\|})$)

Table 2 (continued)

Variable	Units	Default Value	Data Type	Description
Group PLOT2DV (Variables For 2-D Velocity Space Plots)				
IWINDOW(3)		3*0	Integer	Flag for plotting regional window plots (0 = No , 1 = Yes)
ZC2Di *(i = 1 to 3)*	cm	0.0	Real	Center of i-th regional window
ZW2Di *(i = 1 to 3)*	cm	0.0	Real	Half-width of i-th regional window

References

[1] A. Friedman, "An Implicit Particle Mover With Adjustable Damping", Memo PT803002, Lawrence Livermore National Laboratory, Livermore (1988).

[2] B. I. Cohen, A. B. Langdon and A. Friedman, *J. Comput. Phys.*, **46**, 15 (1982).

[3] A. B. Langdon, B. I. Cohen and A. Friedman, *J. Comput. Phys.*, **51**, 107 (1983).

[4] R. J. Procassini, C. K. Birdsall, E. C. Morse and B. I. Cohen, "A Relativistic Monte Carlo Binary Collisional Model For Use In Particle Simulation Codes", Memo UCB/ERL M87/24, Electronics Research Laboratory, University of California, Berkeley (1987).

[5] T. Takizuka and H. Abe, *J. Comput. Phys.*, **25**, 205 (1977).

[6] R. J. Procassini, "Kinetic Particle-In-Cell Simulations Of Transport In A Tokamak Scrape-Off Layer", Ph. D. Thesis, University of California, Berkeley (1990).

[7] I. J. Morey, V. Vahedi and J. Verboncoeur, *Bull. Am. Phys. Soc.*, **34**, 2028, Paper 4U15 (1989).

[8] S. E. Parker, R. J. Procassini, C. K. Birdsall and B. I. Cohen, "A Suitable Boundary Condition for Bounded Plasma Simulation Without Sheath Resolution", submitted for publication in *J. Comput. Phys.*

[9] P. F. DuBois, Z. C. Motteler, P. A. Willmann, *et. al.*, "The Basis System", Report M-225, Lawrence Livermore National Laboratory, Livermore (1990).

[10] B. I. Cohen, A. B. Langdon, D. W. Hewett and R. J. Procassini, *J. Comput. Phys.*, 81, 151 (1989).

[11] R. J. Procassini, C. K. Birdsall and E. C. Morse, "A Fully-Kinetic, Self-Consistent Particle Simulation Model Of The Collisionless Plasma-Sheath Region", submitted for publication in *Phys. Fluids B.*