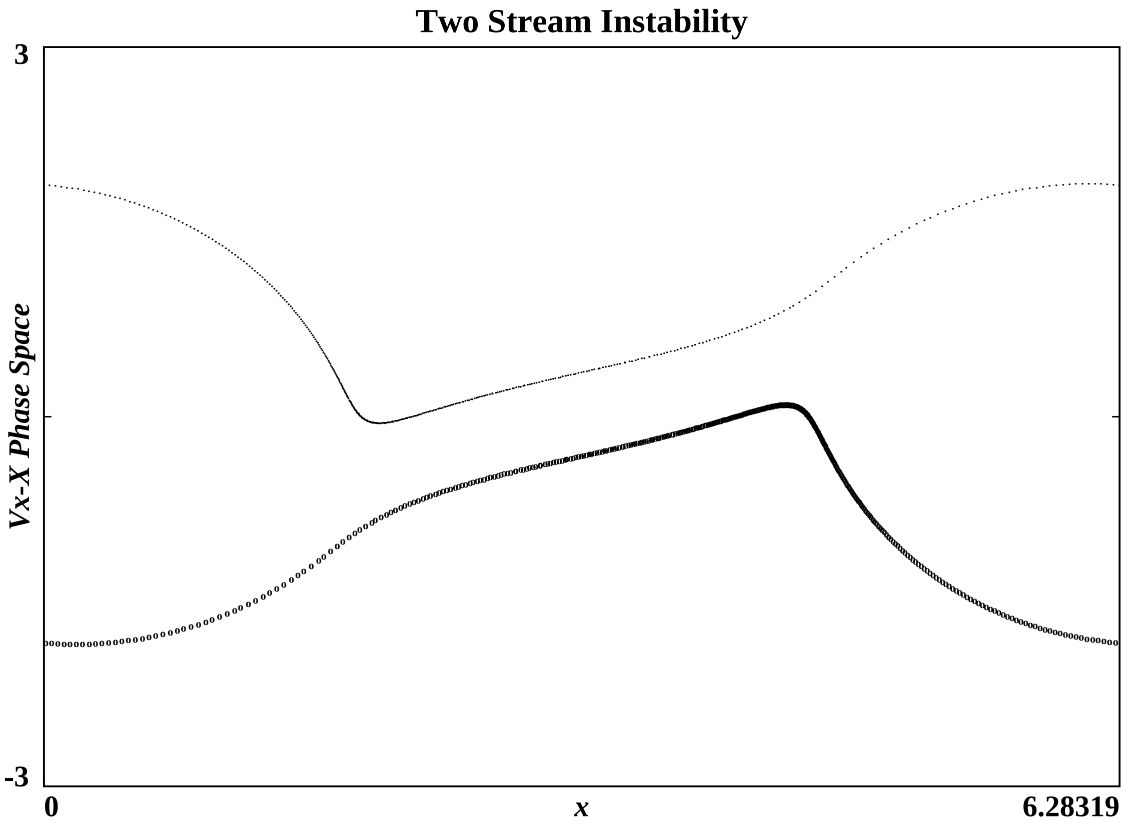
ES1, XES1

ELECTROSTATIC 1 DIMENSIONAL CODE

REFERENCE MANUAL
ES1, XES1 version 4.1



John P. Verboncoeur
Vahid Vahedi
c/o Prof. C. K. Birdsall
Plasma Theory and Simulation Group
Electronics Research Laboratory, Cory Hall
University of California
Berkeley, CA 94720

Contents

1 INTRODUCTION

ES1 is an ElectroStatic 1 dimensional code for the MS-DOS operating system. The code simulates a periodic plasma whose characteristics, including particles and electrostatic fields, are specified by the user at run time using an input file. The simulation proceeds in real-time, with the user viewing output as the code runs in the form of various diagnostics which change with each time step (animation).

The MS-DOS version of ES1 is compiled with the Microsoft C 5.1 Optimizing Compiler. This version of ES1 is almost faithful to the Cray FORTRAN version originated by A. B. Langdon, with the exception of the form of input and output. The PC version provides interactive animation whereas the FORTRAN version produces output files requiring post processing.
In addition, an X-Windows version is also available for Unix workstations implementing the MIT X11R4 standard. The X-Windows versions runs in an X11 environment called XGrafix, with complete source code included. To compile the X-Windows version, the standard MIT X11 libraries are required. The physics code of the X-Windows version is identical to that of the DOS version.

1.1 Scope

This document describes the ES1 program running on the IBM PC and compatibles, as well as the X-Windows version. Program installation, operation, and modification are discussed. In addition, the library of input files accompanying ES1 is described, and the guidelines to generate new input files are provided.

This manual makes no attempt to explain the physics and computational issues of particle simulation. To learn more about particle simulation in general and specifically the physics behind ES1, refer to *Plasma Physics Via Computer Simulation* by C. K. Birdsall and A. B. Langdon (McGraw Hill 1985, Adam-Hilger and IOP 1991). Birdsall and Langdon devote about one third of their text to a detailed discussion of ES1 and its applications.

ES1 is intended as a companion package to that text, although the text is not required to use the program. In either case, some familiarity with plasma physics is required to understand the results of the simulations and generate new simulations. Knowledge of numerical analysis and/or particle simulation is useful for modification of the code and understanding of numerical errors which can occur in any computer simulation.

1.2 Objectives

The objectives of this package include distribution of the ES1 code in a form which makes it accessible to large and small research facilities, universities, and individuals interested in plasma simulation. The IBM PC environment provides a vehicle for distribution since nearly every researcher and student has access to this type of machine. The trend in recent years has been one of decreasing cost and increasing performance and capability.

1.3 History

ES1 was originally written by A. B. Langdon (about 1972) for a course at University of California, Berkeley taught by C. K. Birdsall. The code is both an educational tool and the prototype for many computer simulations. ES1 illustrates many fundamental concepts of plasma simulation, including electrostatic movers, various techniques of particle weighting, noise reduction (smoothing), Maxwellian particle loading, etc. The original FORTRAN code by Langdon has been updated many times; the current version is available to users of the National Energy Research Supercomputer Center at Lawrence Livermore National Laboratory from Langdon. We have also included the source code with this distribution (refer to Section 2.1 Disk Contents for the filename).

ES1 was translated to C in 1985 by T. Lasinski, and ran on IBM XT compatible computers with CGA resolution video. Subsequent modifications are detailed in Section 1.4 Enhancements.

1.4 Enhancements

This section summarizes the enhancements made to the PC version of ES1 since the original version written in Desmet C by T. Lasinski. The enhancements are presented in the order they occurred, referenced by version number. Note that the original version has been given version number 1.0 although it had no explicit version number. There have been a large number of upgrades distributed since we have attempted to respond quickly to requests for specific fixes and improvements.

1.4.1 Version 2.0

This upgrade includes a major conversion of the source code from Desmet CWare compiler to Microsoft C 5.

The graphics have been improved by a factor of nearly 4 in resolution ($320 \times 200 \rightarrow 640 \times 350$ pixels on the EGA) and by a factor of 4 in color (4 colors $\rightarrow$ 16 colors). In addition, the program runs about twice as fast as older versions, and has the capability to use up to 16000 particles, 8192 grid spaces, 8 species, and 1024 time steps, subject to available memory.

The diagnostics have been improved to include electric field, electric potential, kinetic energy history, and total energy history. Also the animation mode we call 'noerase' (see discussion of noerase below) was added.

1.4.2 Version 2.1

The major improvement included in this version is the dynamic allocation of all major arrays. This decreased the size of the executable file from 350K bytes to about 58KB. This fix also enables the program to run small simulations on machines equipped with 256K memory, while handling larger simulations on machines with more memory. Speed is not affected much because the program is loaded from disk faster, but the program must dynamically request memory at run time and initialize the arrays as required.

1.4.3 Version 2.101

The animation screen was enlarged to provide more resolution, and support for IBM CGA in high resolution mode was added.

1.4.4 Version 2.102

An error was discovered in the Maxwellian thermal velocity loaders (used when a non-zero VT1 or VT2 value occurs in the input file). The loaders now are normalized properly and give the correct spread of velocities.

1.4.5 Version 2.103

An error was discovered in the way ES1 rotates velocities in Vx-Vy space for magnetized plasmas. Previously, the rotation was correlated to the spatial position (i.e. the particles were not loaded randomly in velocity space). The particles now are loaded correctly in velocity space.

1.4.6 Version 2.104

General optimizations were performed to streamline much of the code which is called multiple times. The optimizations include use of 8086 register storage for critical variables. The overall improvement in speed is about 10% over previous versions.

1.4.7 Version 2.105

This version adds a perpendicular k-space component to (artificially) in order to simulate two dimensional charge disks. Note that when a value is not specified or $a \to \infty$, ES1 assumes the standard 1d case of infinite charge sheets. Charge disks become infinite charge sheets as $a \gg l$. Also, the charge disks are still really one dimensional as far as the axis of symmetry is concerned - collisions are not avoided in the case of charge disks in a magnetic field because ES1 is still 1d2v. Thus we do not track the y or radial position of particles.

Since ES1 is a one-dimensional simulation, it normally uses infinite charge sheets instead of three-dimensional particles. In PC ES1, we have implemented an optional second dimension to simulate charge disks. The one-dimensional Poisson equation is given by

$$-\nabla^2\phi(k) = \mathrm{K}^2\phi(k) = \frac{\rho(k)}{\epsilon_0} \ ,$$

where

$$\mathrm{K}^2 = k^2\left(\frac{\sin\frac{k\Delta x}{2}}{\frac{k\Delta x}{2}}\right)^2 \ .$$

To simulate charge disks, K is augmented by a perpendicular component:

$$\mathrm{K}^2 = k^2\left(\frac{\sin\frac{k\Delta x}{2}}{\frac{k\Delta x}{2}}\right)^2 + \left(\frac{2.405}{a}\right)^2 \ .$$

Here, a is the finite radius of the charge disk and 2.405 is the first zero of the Bessel function J_0. If a is not specified, infinity is assumed and the charge disks become infinite charge sheets. Note that the while the finite radius of a charge disk affects the potential, particles are still constrained to motion in one dimension (although they can have a perpendicular velocity).
To simulate a background of Boltzmann electrons, using positive ions as the only moving particles, K^2 is augmented by:

$$\mathrm{K}^2 = k^2\left(\frac{\sin\frac{k\Delta x}{2}}{\frac{k\Delta x}{2}}\right)^2 + k_D^2 \ \text{ where } k_D^2 = \left(\frac{n_{e0}e^2}{m\epsilon_0}\right) / \left(\frac{kT_e}{m}\right)$$

The code uses $l/a = l/2.405\lambda_D = 0.4158 l/\lambda_D$.

1.4.8 Version 2.106

In this version we add energy histories for the Fourier-decomposed field energy modes 1, 2, and 3. These diagnostics can be viewed by pressing the number corresponding to the desired mode energy. The display is similar to the field energy history.

1.4.9 Version 2.107

In this version, the display paging scheme (the way animation is accomplished in ES1) was enhanced. The NoErase particle traces are retained even when calling up a diagnostic plot, and screen items are written more efficiently to improve speed.

1.4.10 Version 2.108

With this version, the file extension of .INP is assumed for input files by default. Thus to run the landau damping input file, LANDAU.INP, one could type ES1 LANDAU.INP or equivalently ES1 LANDAU. Note that input files with other extensions can still be run. For example, ES1 LANDAU.NEW would run the input file LANDAU.NEW, while ES1 TEMP would first look for the input file TEMP with no extension, and if TEMP does not exist then ES1 looks for TEMP.INP.

1.4.11 Version 2.109

The input file format was revised to an enable arbitrary number of comment lines between text. The text lines need not be preceded by any specific character - ES1 assumes a line is a text line if it does not contain the next expected data. The lines of numbers are still of the same format; there must be no interceding text between numbers other than spaces or tabs.

1.4.12 Version 2.11

Finally we have semi-log plots for the all energy histories. Now exponential growth rates should be easier to see as they become straight lines on the semi-log plots. The energies are now stored in log10 form, requiring an evaluation of a log for each energy each timestep rather than *nt* evaluations when a time history is requested.

1.4.13 Version 3.0

Version 3.0 is a major upgrade providing ES1 with an improved user interface and output capabilities. The core of the user interface, designed and written by John Verboncoeur and Vahid Vahedi, handles the programming issues of the simulation. These include keyboard handling, screen graphics, and printer output for PostScript and IBM Graphics printers (including the Epson FX compatible family of printers).

The code is separated into a physics application and the windowing core. New physics and diagnostics can be added without altering the windowing code, with the only restriction that any new diagnostic must be a linear, semi-log, or scatter plot. A text plot is currently under consideration which would display parameters from the input file during the simulation.

Using the windowing core, all diagnostics are updated dynamically in time. The core can also update in individual timesteps, pausing for a keystroke before continuing the simulation.

The speed is reduced on the order of 50% compared to previous versions with many plots displayed. The code is faster than previous versions if it is run with no diagnostics since it no longer processes the diagnostic and graphics code.

For magnetized species, ES1 can now display velocity space (v_x versus v_y). This diagnostic is useful for loading ring distributions or observing transverse heating. The diagnostic only appears when at least one species is magnetized (i.e. non-zero cyclotron frequency).

ES1 no longer has a time step limitation; it can run indefinitely. All time histories are combed periodically such that there are never more than 1024 (subject to change) values stored. Note that after long runs this can result in a loss of high frequency resolution on the history plots. This has no effect on the physics of the simulation.

ES1 can now generate hardcopy output on an IBM Graphics Printer or compatible, including the Epson FX series of dot matrix printers. In addition ES1 can generate a PostScript file which can be sent to any PostScript printer or imported into a PostScript document.

1.4.14 Version 3.1

An error was discovered in the species initialization function, init(), in calculating the charge of each species. The charge per unit area, Q, for each species is calculated from the other input parameters, ω_p (the plasma frequency), N (the number of particles), qm (the charge to mass ratio), l (the system length), and $epsi(1/\varepsilon_0)$ to be:

$$Q = l\omega_p^2/(N * qm * epsi) \ .$$

Previously the term *epsi* was not there. Typically *epsi* is set to one for normalized parameters, so its absence was not noticed. However, for running with laboratory parameters, *epsi* must be set to $1/\varepsilon_0$, where ε_0 is the vacuum dielectric constant (or some multiple of it for other materials).

1.4.15 Version 4.0 (XES1)

XES1 is the X-Windows version of ES1 running under X11. This version makes extensive use of a mouse for selecting menus, moving windows, resizing, etc. In addition, XES1 can be run on a remote host while the output displays on a local graphics workstation. XES1 uses XGrafix for the graphics display, which requires X11 libraries to compile.

1.4.16 Version 4.1 (ES1,XES1)

Higher order weightings were added to the move and accel subroutines. The meaning of the iw flag was changed, and a new flag controlling weightings was added. Velocity distribution diagnostics were added; 3 new species parameters and one new global parameter were needed to allow the use of these diagnostics.

2 INSTALLATION

This section describes the contents of the ES1 distribution disks and the installation procedure for hard disk and floppy disk systems.

2.1 Disk Contents

ES1 is distributed on a single 1.44M diskette. This disk contains the files and directories for the IBM compatible PC version as well as the X Windows version.

The ES1 directory contains the required files for the PC version. These are:

ES1.EXE	ES1 executable file. This file must be in the current directory or on the current path to run ES1. See your DOS manual for information on paths and directories.
INSTALL.BAT	Installation program for the PC version to copy all the relevant ES1 files into appropriate directories.
README.TXT	Text file containing information for programmers wishing to modify ES1. When the information in this manual conflicts with the README.TXT file, assume the file is correct. This file may not be present if this manual is up to date with the version of the code distributed.
.C	All files with the C extension are the C language source files for ES1.
.H	All files with the H extension are the C language header files for ES1.
ES1.MAK	The MAKE utility file for automatically performing conditional compilation/linking of only those files which have been changed. Requires the Microsoft MAKE utility provided with many recent compilers (including C 5, Quick C 1, FORTRAN 4, and MASM 5).
MK.BAT	Batch file for automating compilation.

The WIN directory contains the object files for displaying graphics in the PC version. These are:

WINGRAPH.OBJ WINTOOLS.OBJ	Object code (Microsoft format) for thewindowing core used by ES1. Required only if the application will be modified and recompiled. These files are placed in the \WIN directory.
WINGRAPH.H	header file for windowing core features. Required only if the application will be modified and recompiled. This file is placed in the \WIN directory.

The INP directory contains the input files for the PC version. These are:

*.inp All files with the .inp extension are input files. For detailed information on each input file refer to Section 5.2 Input File Library. A directory called inp can set up under the es1 directory (for the PC version) or the xes1 directory (for the X Windows version) to include all the input files.

The FES1 directory contains a tar file, ES1.TAR.Z containing the source files for the FORTRAN version of ES1. The FORTRAN version uses NCAR graphics and has its own documentation.

The DOC directory contains this manual in two formats:

ES1.PS The PostScript version of this document.

ES1.TEX The LaTeX version of this document.

2STREAM.PS A PostScript figure for this document.

The XES1 directory contains a README file and a tar file XES1.TAR.Z which contains the files required for XES1. The README file contains directions for installing XES1 on a Unix platform. Once the compressed tar file is installed, two directories are created: *xes1* and *xgrafix*. The *xes1* directory contains:

*.c All files with the .c extension are the C language source files for XES1. These files should be placed in the xes1 directory.

.h All files with the .h extension are the C language header files for XES1. These files should be placed in the xes1 directory.

makefile The make file for automatically performing conditional compilation/linking of only those files which have been changed. This file should be placed in the xes1 directory.

README.license Contains the license agreement for XES1.

README.upgrade Contains information on how to obtain the latest versions of ES1.

/inp Subdirectory containing the input files for XES1, *.inp. These are the same input files used by the PC version.

/doc Subdirectory containing the documentation files es1.tex (LaTeX version), 2stream.ps, and es1.ps (PostScript version).

The *xgrafix* directory contains the files for XGrafix including the following:

*.c The source files for the XGrafix graphics display library. This file should be placed in the xgrafix directory.

xgrafix.h Header file for XGrafix. This file should be placed in the xgrafix directory.

xgrafix.ico	The XGrafix icon (bitmap). This file should be placed in the xgrafix directory.
xgrafix.str	Another header file containing string definitions for XGrafix. This file should be placed in the xgrafix directory.
Imakefile	The Imakefile file for XGrafix. This file should be placed in the xgrafix directory. If the X libraries are installed properly, run *xmkmf* to generate a Makefile, and then type *make* to create the XGrafix libraries (libXGC.a & libXGF.a). In this case you will not need to use Makefile.xgrafix.
Makefile.xgrafix	The imakefile file for XGrafix. This file should be placed in the xgrafix directory. If you can not find *xmkmf* on your system, modify this file to compile and create the XGrafix libraries.

2.2 Setup and Installation Procedure (PC version)

The setup procedure described in this section assumes the user is familiar with the DOS COPY and MD (Make Directory) commands. The setup of ES1 is similar for single and dual floppy disk drive systems as well as hard disk systems. ES1 is capable of detecting the hardware configuration of the system, including the video adapter, numeric coprocessor, etc., and automatically adjusts to use the hardware to best advantage, so there is no setup procedure required for these items.

If you will be using ES1 on a floppy-based system, backup the ES1 distribution disk using the DOS DISKCOPY program. Do not remove the write protect tab on the original ES1 diskette to ensure that you retain a copy of the original, unmodified executable code, source code, and input files. This completes the installation procedure for floppy disk systems.

For automated installation to any type of disk, make the floppy disk containing the ES1 files the current disk by entering A: at the DOS prompt. Enter INSTALL C: to copy the files to drive C: (replace 'C' by the appropriate drive letter if you have more than one logical hard disk drive). This program will create a directory \ES1 and copy the ES1 files to it, and will also create a directory \WIN and copy the object and header files for the windowing core. Note that the contents of the \WIN directory are only required if the code will be modified.

To use ES1 in these configurations, you must first make the ES1 directory the current directory by typing CD ES1. Then follow the directions below for program operation.

2.3 Setup and Installation Procedure (X Windows version)

The installation procedure for the X Windows version must be done manually. Only the contents of the XES1 directory are needed for this version. Place the contents of the XES1 directory in the home directory on the Unix machine. Follow the directions shown in the file README to extract ES1.TAR and install files in the appropriate directories.

3 MS-DOS PROGRAM OPERATION

3.1 Syntax

ES1 INP\filename[.inp],

where < filename.inp > is the name of the input file. Although we have used *.INP for the input files in the library, the .INP extension is not required. If no filename is provided on the command line, ES1 looks for a file named ES1DATA (to maintain compatibility with previous versions). If the input files are not in the same directory or are located in a sub-directory, the path must also be specified. For instance, the syntax for starting ES1 with the input file 2stream.inp which is in a sub-directory of es1 called INP is:

ES1 INP\2STREAM[.INP]

The input file is required since ES1 determines the parameters of the simulation at run time. The input parameters are displayed and the computer pauses for a keystroke.

3.2 Run, Stop, and stEp

ES1 is a real-time simulation; you can view the results of the simulation as they occur. When the simulation is paused (i.e. time is not being incremented), there is a menu option **Run** available. When **Run** has been selected, it is replaced on the menu by **Stop**. Thus to initiate or continue the simulation, select **Run**. To pause the simulation, select **Stop**. Note that ES1 is not limited in the number of timesteps since time histories are combed periodically. The timestep counter may become negative after 32k timesteps, but this affects only the displayed number; internally the physics are still consistent, and the timestep will continue to increment, eventually becoming positive again.

The **stEp** option causes ES1 to take a single timestep and pause. This option is available at all times. **stEp** is useful for viewing discrete temporal changes and studying a rapidly changing simulation in detail before the phenomena of interest (an instability, for example) is completed. When the simulation is running (after selecting the **Run** option), **stEp** has the affect of stopping the simulation and then taking a single step, leaving the simulation paused (**Run** appears on the menu and can be selected).

3.3 Tools

This menu item provides the tools for modifying the diagnostics windows. The services provided by **Tools** include moving windows to a new location, changing the size of the windows, changing the plotting limits for the windows (including an automatic rescale), and turning on noerase for a window. All features provide interactive directions while in use.

3.3.1 Move

The **Move** feature enables windows to be moved to a new location on the display. After selecting **Move**, the list of diagnostics is presented. To select a window, move the highlight bar to the appropriate item and press Enter or press the first letter of the desired item. If the window is currently on the screen, the outline is highlighted while the user drags the window to the new location using the cursor keys. If the window is not currently active, its outline appears and it is made active once the **Move** operation is completed. The cursor keys move the outline in 1 pixel increments, while the cursor keys in conjunction with shift move the outline in 8 pixel increments. When the diagnostic has been moved to the desired location, press $\boxed{\text{Enter}}$ to complete the **Move** operation.

3.3.2 Resize

The **Resize** feature provides the capability to alter the size of a displayed window. To change the size, select **Tools, Resize**. The list of available diagnostics is displayed. After selecting a diagnostic, ES1 prompts for a window edge to resize. Select the edge using the cursor keys (or shift-cursor keys). The window outline is highlighted, and the selected edge can be resized in either direction. When the edge has been moved to the desired size, press $\boxed{\text{Enter}}$. At this point, the user is prompted to select another side to resize. Continue to select edges to resize until the desired size is obtained. Pressing $\boxed{\text{Enter}}$ at the select edge prompt completes the operation.

3.3.3 rEscale

To change the displayed limits of a plot select **rEscale**. A list of available diagnostics is displayed. This function only works on active diagnostics. After selecting an active diagnostic, a dialog box containing the scaling information for the window is displayed. The items displayed include the upper and lower limit of the x- and y- axis, the Autorescale toggle for each axis, and the OK and CANCEL buttons. Use the cursor keys to move the highlight to the desired item. To change a number, enter the new number and press $\boxed{\text{Enter}}$ or $\boxed{\downarrow}$. To toggle Autorescale, press any key. To apply the changes made, select the OK button and press $\boxed{\text{Enter}}$. To cancel changes, select CANCEL and press $\boxed{\text{Enter}}$ or press $\boxed{\text{Esc}}$.

The x- and y- axis labels do not affect the simulation; only the portion of the plot which can be seen is affected. Autorescale causes the windowing core to update the axis labels each time step based on the maximum and minimum values. For linear plots, the maximum or minimum is found each time step. For semi-log plots, autorescale will only increment the labels in decades. To ensure that the plot does not span too many decades due to a wide variation on a log scale (such as a quantity which goes nearly to 0), autorescale selects at most 6 decades to plot starting from the maximum value.

Note that autorescale now works correctly on plots with a multiple items; i.e., a phase space plot with several species can now be autorescaled.

3.3.4 Noerase

The **Noerase** item provides the capability to leave the plot at the previous timestep on the screen when drawing the new plot. This is often useful for watching the trajectories of particles in phase space to determine trapping regions or holes in phase space, or to observe the transient behavior of a snapshot for a brief time.

When **Noerase** is selected, the list of available diagnostics is displayed. The operation is completed by selecting a diagnostic window to apply the noerase feature to.

Note that there are times when the noerase plot cannot be saved. This includes cases when another window overlaps the window with noerase on, as well as resizing, moving or rescaling the diagnostic.

3.3.5 Cross-hair

The **Cross-hair** tool provides the user with a pointer to obtain the coordinates of any point in any window (in coordinate system of the window e.g. semilog, etc.). The coordinates are displayed in the upper left corner of the screen.

When **Cross-hair** is selected, the simulation is paused. Pressing cursor keys will move the cross-hair in vertical and horizontal directions. To exit and resume simulation press $\boxed{\text{Esc}}$ or $\boxed{\text{Enter}}$.

3.4 Diagnostics

ES1 provides diagnostics for many simulation parameters of interest. Diagnostics can be viewed at any time and in any arrangement desired. The diagnostic windows can be resized, moved, rescaled, and closed. Note that the program speed is affected slightly by the number of diagnostics displayed since each diagnostic requires some additional processing each timestep when it is open.

Diagnostics are opened or closed in the same manner; the **Diagnostic, <menu item>** sequence is a toggle which opens the diagnostic if previously inactive. If the diagnostic is currently active, it will have a marker next to its name in the list of diagnostics. Selecting an active diagnostic makes it inactive.

3.4.1 VX Phase Space

The VX Phase Space diagnostic contains the menu bar as well as the plot of velocity versus position for all particles of all species. Particles of different species are assigned different colors to distinguish them. Often different species are used for particles of different characteristics such as charge, mass, initial velocity, etc. Note that since ES1 is a periodic code, particles which exit one side of the Phase Space plot will return through the other side with the same velocity. Although ES1 processes velocities and positions in normalized form, the plots are denormalized to reflect values in the same units as the input parameters in the input file.

VX Phase Space is displayed by selecting **Diagnostics, VX Phase Space** from the menu.

3.4.2 Electric Field

The Electric Field diagnostic displays the value of the electric field at the grid points. Since the electric field is the spatial integral of the density, it is smoother. ES1 uses the electric field to determine the electrostatic force on particles in the mover. The electric field is also periodic.

Electric Field is displayed by selecting **Diagnostics, Electric Field** from the menu.

3.4.3 Potential

The Potential diagnostic displays the value of the potential at the grid points. Since the potential is the spatial double integral of the density, it contains even less noise than the electric field. The potential is also periodic.

Potential is displayed by selecting **Diagnostics, Potential** from the menu.

3.4.4 Density

The Density diagnostic displays the space charge density of all particles as it is weighted to the grid. Note that the density is periodic; i.e., $\rho[0] = \rho[NG]$. The density may be noisy due to the small number of particles being weighted to a finite grid.

Density is displayed by selecting **Diagnostics, Density** from the menu.

3.4.5 Kinetic Energy

The Kinetic Energy diagnostic displays the time history of the total kinetic energy for all species. This history is a semi-log plot.

To display the Kinetic Energy history, select **Diagnostics, Kinetic Energy** from the menu bar.

3.4.6 Total Energy

The Total Energy diagnostic displays the time history of the total energy of all species; i.e. total energy = kinetic + field energies. Note that this history is a semi-log plot, calculated from

$$\log(TE_i) = \log(KE_i + ESE_i) \ .$$

The total energy of a given simulation should be conserved; numerical error is indicated when the total energy changes. In simulations where the total energy changes by more than a few percent, there is serious error in the solution possibly resulting from a timestep exceeding the limit:

$$\omega \Delta t \leq 2 \ .$$

Here ω is the highest frequency in the simulation. Exceeding the limit results in exponential growth and damping in the leapfrog movers. See Birdsall and Langdon for a detailed analysis.

To display the Total Energy diagnostic, select **Diagnostics, Total Energy** from the menu.

3.4.7 Field Energy

The Field Energy diagnostic displays the time history of the electrostatic field energy. Note that this is a semi-log plot, enabling the viewer to determine exponential growth/damping rates of energy more easily (e.g., the growth rate of an instability).

Field Energy is displayed by selecting **Diagnostics, Field Energy** from the menu.

3.4.7.1 Fourier Modes

The Fourier modes of the electrostatic energy, Mode 1 ESE, Mode 2 ESE, etc., display the respective mode of the Fourier spatially decomposed electrostatic energy. These history plots are semi-log, emphasizing exponential growth rates. Modes up to NG/2 are available from the physics of the simulation; the displayable modes are set by MMAX in the input file.

To display the fundamental mode energy history, select **Diagnostics, Mode 1 ESE** from the menu. To display the second mode energy history, select **Diagnostics, Mode 2 ESE** from the menu. To display the n*th* mode energy, select **Diagnostics, Mode** *n* **ESE** from the menu.

3.4.8 Velocity Space

The Velocity Space diagnostic displays a scatter plot of the v_y versus v_x velocity space of magnetized particles only. Note that v_y is identically zero for unmagnetized species since ES1 is a 1d2v code (1 spatial dimension, 2 velocity components). The scale is the same for both axes.

Velocity Space is displayed by selecting **Diagnostics, Velocity Space** from the menu.

3.5 Print

The **Print** menu item provides printing capability for the application. Currently supported are two types of printing: a screen dump to a dot matrix printer (IBM Graphics printer or compatibles, including the Epson FX series), or a PostScript dump to a file for later printing on a PostScript laser printer.

3.5.1 PostScript

This option generates a high resolution (300 dots per inch on most laser printers) vector plot of the currently displayed diagnostics on the screen. The vector plot is stored in an ASCII file in the PostScript page description language for later printing and/or inclusion in PostScript documentation. Note that the file can become quite large (up to several hundred k bytes) if phase space/velocity space is plotted for a large number of particles. Linear and semilog plots do not add much overhead to the file.

The plot corresponds closely to the screen display with higher resolution. Colors are generally translated into gray shades, with the exception of scatter plots, which translate colors into different symbols so the various species can be distinguished.

When the **PostScript** item is selected, a dialog box is displayed. The items contained on the dialog box include the filename to dump into, the dump period, and the dump limit.

The default filename is *output.ps*. To change to a new filename, move the highlight bar to the filename field and enter a new filename. If the file already exists from a previous session, it is replaced by the new file. If the file was previously created during the current session, the new plot is appended to it. Note that the file is placed in the current directory (normally the \ES1 directory).

The dump period is the number of timesteps which pass between generation of plots. If the dump period is set to 100 at timestep 100, a plot is generated for timestep 100, 200, 300, ... until the dump period is changed or the maximum number (see dump limit) of plots have been generated. To dump only at the current time regardless of the setting of dump limit, use 0 for dump period.

The dump limit is the maximum number of plots which are to be generated. This feature is used in conjunction with dump period to create a file containing several plots spaced an equal number of timesteps apart.

When all desired options have been set, move the highlight bar to OK and press $\boxed{\text{Enter}}$. To exit without plotting to the file and restoring old settings, move the highlight bar to CANCEL and press $\boxed{\text{Enter}}$.

3.5.2 CGM

This option generates a high resolution (16-bit integer) vector plot of the currently displayed diagnostics on the screen. The vector plot is stored in a binary file conforming to a subset of the ANSI CGM Standard Version 1. The CGM file can be read by many graphics programs for later editing, printing, and/or inclusion in documentation. Note that CGM files are more compact than the equivalent PostScript file, but may still become large if phase space/velocity space is plotted for a large number of particles. Linear and semilog plots do not add much overhead to the file.

The plot corresponds closely to the screen display with higher resolution. Colors are generally translated into gray shades, with the exception of scatter plots, which translate colors into different symbols so the various species can be distinguished.

When the **CGM** item is selected, a dialog box is displayed. The items contained on the dialog box include the filename to dump into, the dump period, and the dump limit.

The default filename is *output.cgm*. To change to a new filename, move the highlight bar to the filename field and enter a new filename. If the file already exists from a previous session, it is replaced by the new file. If the file was previously created during the current session, the new plot is appended to it. Note that the file is placed in the current directory (normally the \ES1 directory). If multiple files are to be generated (dump period > 0, dump limit > 1), they are named in sequence by appending a number to the end of the filename. Some characters at the end of long filenames may be removed to fit the plot number. This is necessary since CGM defines only a single page per file.

The dump period is the number of timesteps which pass between generation of plots. If the dump period is set to 100 at timestep 100, a plot is generated for timestep 100, 200, 300, ... until the dump period is changed or the maximum number (see dump limit) of plots have been generated. To dump only at the current time regardless of the setting of dump limit, use 0 for dump period.

The dump limit is the maximum number of plots which are to be generated. This feature is used in conjunction with dump period to create a file containing several plots spaced an equal number of timesteps apart.

When all desired options have been set, move the highlight bar to OK and press $\boxed{\text{Enter}}$. To exit without plotting to the file and restoring old settings, move the highlight bar to CANCEL and press $\boxed{\text{Enter}}$.

3.5.3 Dot Matrix

This option generates a screen dump to the printer at the resolution of the screen. The screen rotated 90 degrees to best fit on a sheet of paper. Note that the resolution resulting on the printer in this case is limited to that of the screen. Dumping to a dot matrix printer may take up to one minute depending on the speed of the printer.

3.5.4 Dump

This option is NOT implemented in this version.

3.6 Quit

The quit option ends the simulation, exiting back to the DOS prompt. To avoid accidentally selecting quit and loosing the data of a lengthy simulation, the user is prompted to ensure that an exit is desired. Responding $\boxed{\text{OK}}$ will exit, $\boxed{\text{CANCEL}}$ will resume the simulation.

3.7 Shortcut Keys

The menu selections on all menus are chosen either by moving the highlight bar to the desired item and pressing $\boxed{\text{Enter}}$, or pressing the capitalized letter in the menu item. For example, to select **stEp**, the user could press $\boxed{\text{e}}$ or move the highlight bar to **stEp** and press $\boxed{\text{Enter}}$. Note that any unrecognized keystroke will generate a highlight bar if one did not exist on the menu, or will be ignored if a highlight bar is currently displayed. The menu manager is insensitive to case; pressing $\boxed{\text{E}}$ in the previous example is equivalent to $\boxed{\text{e}}$.

Items in the **Diagnostics** list are selected by pressing the letter corresponding to the first character in the name. In the event of conflicting first characters, the first matching item is selected.

4 X-WINDOWS PROGRAM OPERATION

The X-Windows version of ES1, XES1, is operated in the same manner as discussed in the previous section, with the exceptions noted below.

4.1 GUI Support

XES1 fully supports a mouse for selection of items, buttons etc. Moving, resizing, and iconifying of windows is supported indirectly via the X window manager (Motif, Open Look, etc.). Keystrokes are not supported for these actions, so a mouse is required. The move, resize, and iconifying buttons and operations are governed by the window manager; consult the window manager manual or guru for details of these procedures.

4.2 Main Menu

The buttons on the main menu can be selected using the mouse. The functions available include RUN, STOP, STEP, SAVE, and QUIT, which all perform the same function described previously in Section 3. Note that the SAVE function is equivalent to the DUMP function in the MS-DOS version which is also NOT implemented in this version.

4.3 Diagnostic Window Buttons

Every diagnostic window in XES1 contains four buttons: Rescale, Trace, Print, and Crosshair.

4.3.1 Rescale

The rescale button pauses the simulation and opens a dialog box containing editable fields for the minimum and maximum labels on the x and y axes. In addition, the dialog box contains buttons for automatic rescaling of the x and y axis. These buttons toggle autorescaling of the respective axis on and off. When all axes are scaled as desired, select OK to accept the changes or CANCEL to return to the previous status. Note that while rescaling the simulation is paused.

4.3.2 Trace

The trace button turns toggles the plot tracing feature on and off. The previous plots are accumulated, generating a series of lines or dots as described above.

4.3.3 Print

The Print button generates a PostScript plot file of the current window. Pressing the button opens a dialog box containing the file name for the plot and a plot title. Selecting OK generates the plot, CANCEL returns to the simulation. Note that the simulation is paused while the dialog box is open.

4.3.4 Crosshair

The crosshair button activates the crosshair pointer and opens a dialog box displaying the coordinates of the pointer. To display the coordinates of a point move the crosshair pointer to the desired location and click. The simulation is paused until the crosshair is deactivated by selecting the Crosshair button again.

4.4 Diagnostics

In XES1 diagnostics there is no diagnostic menu list. Instead, all diagnostics appear as icons at the bottom of the display or in the designated icon area depending on the window manager. To open a diagnostic, simply click on its icon. In addition, some window managers will display a list of the available diagnostics with all other open windows in a window list. Note that this is not a feature of XES1, but rather a feature of the X-Windows Manager used on the system.

5 CASE STUDIES

ES1 obtains its versatility through the use of input files. The input file contains the parameters for the simulation, specifying number of each species, number of grids, etc. This section describes the contents, use, and modification of input files for ES1.

5.1 Input File Parameters

The codes use input files to describe the simulation with global parameters, as well as the parameters describing each species of particles. The parameters are also described in B&L (Sect. 3-3).

5.1.1 Global Parameters

nsp	The number of particle species to simulate (0= no species present, 1= one species in the whole system, etc.). If modifying an input file that has, say, 2 species, to add more species, just copy one of the blocks of parameters corresponding to species 1 or 2, and change the parameters to the desired values. Note that each species added seeks another 100 k Byte of memory.
l	The length of the system.
dt	The time step.
nt	The total number of steps to be run.
mmax	Maximum number of electrostatic energy modes to be view. Note that this parameter is only for diagnostic purposes and does not enter the calculations.
l/a	See Section **1.4.7** of this manual.
ng	The total number of grid points (power of 2).
iw	Weighting to be used: 0 for zero order(NGP) 1 for first order (CIC, PIC) 2 for second order (quadratic spline) 3 for third order (cubic spline)
ec	Momentum conserving/Energy conserving flag 0 Momentum conserving scheme (recommended) 1 Energy conserving scheme The momentum conserving scheme uses the same weighting for particles and forces. The energy conserving scheme uses a weighting of one lower for forces than is used for particles.
epsi	$1/\epsilon_0$ (usually 1).
a1	Compensation factor (a1 = 0 means no compensation).
a2	Smoothing factor (a2 = 0 means no smoothing).

E0 Magnitude of an applied electric field.

w0 Frequency of the applied electric field.

accum Velocity diagnostics parameter. 0 turns them off
Other positive integers determine the number of timesteps to accumulate the diagnostics.

5.1.2 Species Parameters

One set for each species should be specified.

n Number of particles.

nv2 Exponent of quiet start distribution $f(v) \propto (v/v_{t2})^{nv2}\exp(-v^2/2v_{t2}^2)$, usually zero.

nlg Number of sub groups to be given the same velocity distribution, usually one or ng.

mode Number of mode to be given an initial perturbation in x, v_x.

wp ω_p (positive).

wc ω_c (signed).

qm q/m (signed).

vt1 Provides Gaussian velocity distribution of thermal velocity v_{t1} centered on $v_x = v_0$, $v_y = 0$, using random number routine; maximum velocity is $6v_{t1}$.

vt2 Provides Gaussian (or other) velocity distribution of thermal velocity v_{t2} using inverse distribution functions, giving ordered velocities ("quiet start").

v0 Drift velocity in x direction (signed).

x1 Magnitude of perturbation in x, generally less than half the uniform particle spacing, n/l; used as $x1\ \cos(2\pi x mode/l + \theta_x)$

v1 Magnitude of perturbation in v; used as $v1\ \sin(2\pi x mode/l + \theta_v)$

thetax θ_x

thetav θ_v

nbins The number of bins to use when accumulating the velocity distrubution diagnostics. 100 works well.

vlower The lower limit of the velocity diagnostic for this species.

vupper The upper limit of the velocity diagnostic for this species.

5.2 Input File Library

Clearly the number of data sets is virtually unlimited. ES1 is accompanied by a number of prepared simulations, many corresponding to the projects discussed in Birdsall and Langdon. The user is encouraged to make a working copy of the data sets to edit using any ASCII word processor or editor. Note that the relative position of the numeric data is important for useful results, but the number of spaces separating each number is unimportant as long as the numbers remain on the same line. Comments may be added on the lines containing the descriptive text (so long as it remains a single line) and at the end of the input file an unlimited number of text lines are supported.

5.2.1 2STREAM.INP

The two stream instability is far more instructive on the PC since we can observe the growth and streaming take place rather than viewing intermittent phase plots (a la Cray output). In addition, we use a different color for each species so the species can be tracked as they warm up. This is a good input file to try the NoErase option on to find the trapping regions. Can observe growth of the instability (and eventual saturation) using energy histories. See Birdsall and Langdon, Secs. 5-8 and 5-9.

5.2.2 3STREAM.INP

Similar to 2STREAM, except we add a cold non-drifting species. Also try the NoErase option to observe the trapping regions.

5.2.3 4E4E.INP

Similar to a two stream instability, we can look at just 4 particles to observe streaming, trapping, etc. Use the NoErase option to view trapping regions.

5.2.4 4E4P.INP

We try to form atoms with 8 particles, and observe that when an atom-pair is beginning to stabilize, another particle tends to interfere and break the pair up.

5.2.5 4STREAM.INP

Similar to 2STREAM, we use 4 streams to observe a double two stream instability. Also try NoErase to observe the trapping regions.

5.2.6 BEAMCYC.INP

Beam cyclotron instability; can observe growth using energy histories. See Birdsall and Langdon Sec. 5-14.

5.2.7 BEAMPLAS.INP

Beam plasma instability; can observe growth using energy histories. See Birdsall and Langdon Sec. 5-12.

5.2.8 COLD17.INP

Similar to COLDPLAS, except we perturb in mode 17 with NG=32 to observe aliasing back to mode 15 in density. Also try perturbing in mode 32 to get equivalent of a mode 0 perturbation - nothing happens. See Birdsall and Langdon page 92.

5.2.9 COLDBEAM.INP

Cold beam numerical instability; can observe aliasing effects using phase space plot and any grid diagnostic. Can observe growth of the instability using energy histories. See Birdsall and Langdon Sec. 8-12.

5.2.10 COLDPLAS.INP

Cold plasma oscillations similar to TWOOSC except we add many particles (this is a mode 1 perturbation). See Birdsall and Langdon Sec. 5-4.

5.2.11 EE.INP

Similar to 4E4E, we can observe the two stream instability with 8 particles.

5.2.12 EP.INP

Loading an electron and positron of equal masses and opposite drift, we form a 'atom' (this is not clearly observable on batch machines with intermittent plots). Try the NoErase option (Trace in the X-Windows version) to observe the trajectory of the particles in time.

5.2.13 EPMASS.INP

Similar to EP except the positive particle is 1000 times more massive than the negative particle. We end up with a translating atom with the electron vibrating about the proton.

5.2.14 HYBRID.INP

Hybrid oscillations, similar to those described in Sec. 5-5 in Birdsall and Langdon. The oscillations are similar to cold plasma oscillations, except the plasma is in a magnetic field which changes the frequency of oscillation to the hybrid frequency. See Birdsall and Langdon Sec. 5-5.

5.2.15 LANDAU.INP

Landau damping using only 2048 particles. This one may be a little slow on a 4 MHz PC; it takes about 1.5 seconds per timestep on an 8 MHz IBM AT. See Birdsall and Langdon Sec. 5-15.

5.2.16 LANDAUP.INP

Using LANDAU.INP with the excitation of mode 1 at $k\lambda_D = 0.4$, it is found that $\omega_{real} = 1.2\omega_p$, which makes the wave phase velocity $\omega_{real}/k = 1.2$. In order to observe the phase space in the frame of the wave, shift the Maxwellian by v_{phase} by using $v_0 = -1.2$. Now the $v_x = 0$ axis is the wave frame. At $t = 0$ expand the v_x axes to +- 0.2, turn on the trace, and run the simulation. Observe the very few particles near $v = 0$ escape the wave trap and become passing particles, as the wave damps. Next, reduce the initial excitation until the Landau damping rate reduces to that in texts, and observe the phase space change from trapping to phase mixing. Next increase the initial excitation until trapping dominates.

5.2.17 TWOOSC.INP

Two particle cold oscillations. This is more instructive to run on the PC than on the Cray since one can observe particles bouncing in simple harmonic motion. A good demonstration of ES1 movers is to increase V1 so the particles pass through each other without elastic collision to convince yourself that ES1 avoids singularity in fields.

5.3 Format

The input file is currently a fixed format ASCII file. The input files may be edited using any ASCII text editor or word processor which does not insert formatting characters into the file. The text lines can contain any descriptive comments, etc., and the text may continue for as many lines as desired. The lines containing numbers must also remain on a single line and each number can only be separated by whitespace, including any amount of spaces and tabs. The numbers may be in floating point or exponential format (decimal point and sign are optional).

The preferred method of trying new input parameters is to COPY the original data set to a new file name and edit the working copy. This will leave the original data set intact for future use and reference.

The syntax of the COPY command is given in the DOS manual. For example, to edit the Landau Damping input file, first type COPY LANDAU.INP LANDAU1.INP to create a copy of the original data. Now the copy, LANDAU1.INP, can be edited using the instructions included with your editor.

A APPENDIX SYSTEM REQUIREMENTS

The minimum MSDOS hardware configuration required to run ES1 is as follows:

- IBM XT, AT, PS/2, or compatible computer. (80286 or 80386 recommended)
- DOS version 3.0 and above.
- 512K system memory (640K recommended).
- IBM-compatible EGA, or VGA color graphics adapter (EGA or VGA and color monitor required).
- 8087/80x87 numeric coprocessor recommended (provides increase in speed of over an order of magnitude).

In addition, a 32 bit 80386 or 80486-based processor running at a high clock speed is recommended for increased performance.

The X-Windows version requires X11 libraries (X11R4 or any superset of X11 such as Motif), a C compiler, and an X display or X-terminal.

B APPENDIX TECHNICAL SUPPORT

Current users interested in new versions should contact us periodically; we currently are not staffed to notify our entire user base. Technical questions, comments, and suggestions on this software and documentation can be sent to:
birdsall@eecs.berkeley.edu

Prof. C. K. Birdsall
EECS Dept., Cory Hall
University of California
Berkeley, CA 94720