Flying Ions

Introduction

This chapter covers how to fly ions in SIMION's ion optics workbench. All Ions are flown within the **View** function. The material below assumes that you know how to create potential arrays, refine them, and project instances of them into a workbench volume. It is assumed that you have read the discussions of potential arrays in Chapter 2, 4, 5, 6, and 7. If not, read the material before proceeding further.

Chapter Organization

This chapter covers ion flying in the following order:

- Defining the ions to fly.
- Flying the defined ions.
- Selectively recording ion flight data.
- An introduction to user programs.

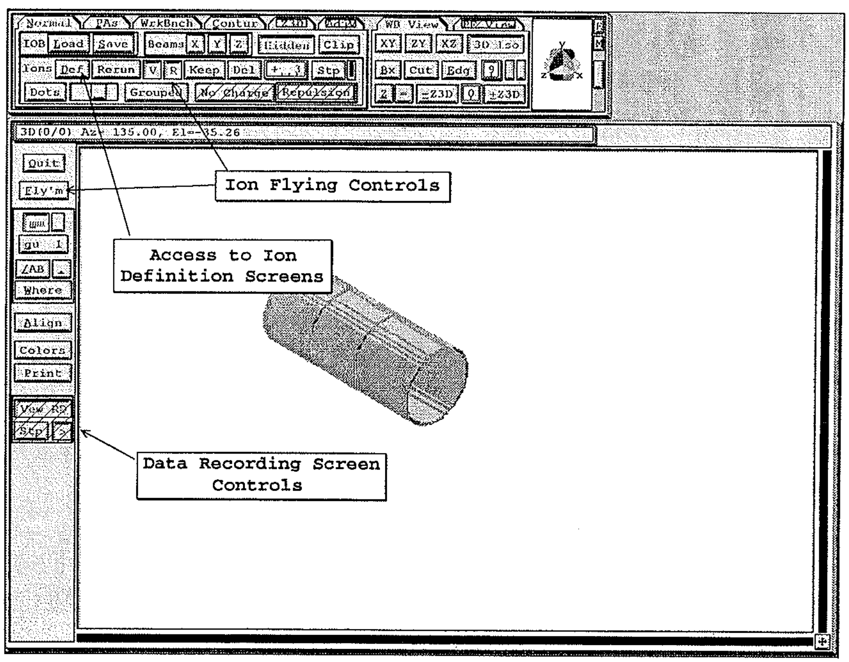


Figure 8-1 Location of various ion definition, flying, and recording controls on the View Screen

SIMION's Unit System

SIMION uses the following ***internal*** unit system for all its ion flying:

Mass	***Unified*** atomic mass units (*amu*).
Charge	Elementary charge units (*e.g. electron has a charge of -1.0*).
Energy	Electron Volts (*eV*).
Time	Microseconds from the start of the **Fly'm** clock.
Distance	Millimeters in the ***currently aligned*** ***workbench coordinates***.
Velocity	Millimeters/microsecond in the ***currently aligned*** ***workbench coordinates***. This is the same as ***meters/millisecond*** or ***km/second***.

Defining the Ions to Fly

The Normal Controls Screen, in **View**, contains a collection of ion flying controls (*Figure 8-1 above*). The **Def** button is used to access the Ion Definition Screen.

Ion Definition Parameters

SIMION uses a collection of parameters to define each ion. The following material discusses each parameter and its use:

Ion's Mass (amu)

The mass of the ion is defined in ***unified*** atomic mass units (*amu*). This is the ***rest*** mass of the ion.

Ion's Charge (elementary charge units)

The ion's charge is specified in elementary charge units. For example, the charge on an electron is -1.0 elementary charge units.

Ion's Starting Kinetic Energy (eV)

The ion's starting kinetic energy is specified in electron volts (*eV*). SIMION translates this starting kinetic energy into initial ion velocity (*mm/microsecond*) using the ion's mass and applying relativistic corrections as required. This velocity will initially be in the ion's starting direction (*defined below*).

Ion's Starting Location

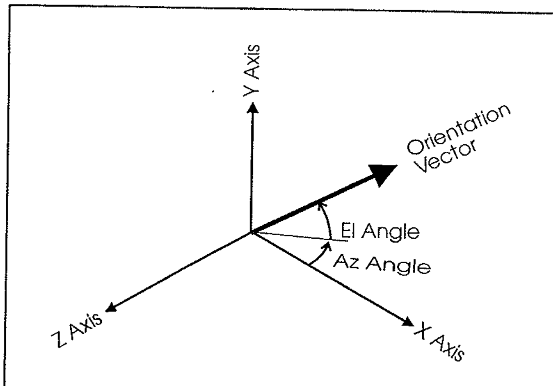
This is the x, y, and z coordinates of the location where the ion first appears. SIMION ***always*** uses (*internally*) the ***currently aligned*** ***workbench coordinates*** for ion locations. However, you must ***define*** the ion's starting location in one of the following coordinate systems (***the same coordinate system for all ions defined***):

1. ***Unaligned*** ***workbench coordinates*** (*mm*). These are independent of the current status of the **Align** button (*handy*).
2. In grid units (*gu*) from the ***working origin*** of a user specified instance.

SIMION will ***automatically*** convert ion starting locations defined in any of the above coordinate systems into the ***currently aligned*** ***workbench coordinates*** for its internal use.

Ion's Starting Direction

The ion's starting direction is defined through the use of azimuth (*Az*) and elevation (*El*) angles (*in degrees*). *Note: El is applied first then Az*. These angles are ***relative*** to the coordinate axis system used to define the ion's starting location (*diagram on right*). For example, if the ion's starting location was specified in instance number one's coordinates, the orientation angles would be relative to instance number one's x-axis, y-axis, and z-axis.



Angular Orientations

Ion's Time of Birth (TOB)

SIMION starts the elapsed time clock when you click the **Fly'm** button to start flying the ions (*units of time are in microseconds*). Normally all ions are created at time equals zero. However, you have the option of a delayed time of birth for each ion. The time of birth parameter specifies this ***delay*** in microseconds.

Ion's Color

SIMION allows individual ions to have different colors. Any of SIMION's 16 colors (*color index 0-15*) can be selected.

Note: Selecting a color of 15 will make the ion invisible. This is useful if you are using charge repulsion on a group of ions and only want to see the trajectories of one or more selected ions to keep down the clutter.

Ion's Charge Weighting Factor (CWF)

SIMION 7.0 supports certain types of charge repulsion. To speed calculations, each ion normally represents a cloud of ions. However, the exact relative number of ions that each ion represents may be different. For example, if each ion represents the ions in rings of cylindrical slices, and the ***actual*** ions are uniformly distributed, the ions in the outer rings probably represent more physical ions than the ions in the inner rings.

The charge weighting factor (*CWF*) is provided to allow you to tell SIMION that this ion represents 50% more actual ions (*CWF = 1.5*) than the normal ion (*CWF = 1.0 - default value*).

The Top Row of Controls on the Ion Definition Screen

The top row of controls perform the following functions (*Figure 8-2 below*):

The Cancel Button

The **Cancel** button discards any ion definition change and restores the ion definition active when the **Define** button (*on the Normal Controls Screen*) was clicked.

The OK button

The **OK** button exits the Ion Definition Screen and keeps any ion definition changes as the current ion definitions.

The Coordinate System Selector

The **Coordinate System** selector is used to select the coordinates to use for locating the ions. The options are: ***Unaligned*** workbench coordinates or instance coordinates of ***one*** of the currently defined instances.

In Figure 8-2 (*below*) the coordinates chosen are instance number one's coordinates. The ***second*** number one (*1- to the right*) indicates that there is ***only one*** instance defined in the current workbench. Thus the ion starting locations are in grid units (*gu*) from instance number one's ***working origin***. Also, the orientation angles are relative to array instance number one's x, y, and z axis.

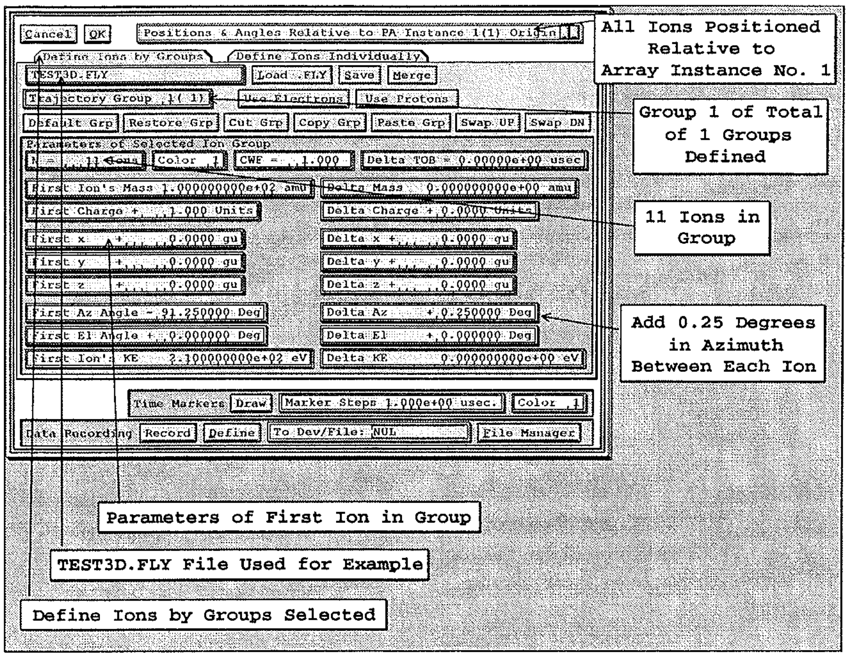


Figure 8-2 Example of the Ion Definition Screen

How Ions Are Defined

Ions are defined either in ***groups*** of similar ions or ***individually***. Each ion definition method has advantages for certain types of simulations. SIMION provides access to either definition method via tabs on the Ion Definition Screen (*Figure 8-2*).

Defining Ions by Groups

Clicking **Define Ions by Groups** tab on the Ion Definition Screen accesses the Group Definition Screen (*Figure 8-2 above*). This definition screen allows you to define ions in groups of ***similar*** ions. Up to **25** of these groups can be defined at one time.

A group of ions is defined by a starting ion definition (*ion number **one** in the group*) and ***parameter increments*** for use in defining the additional ions in the group. For example, in Figure 8-2, there is one group of ions defined. This group has 11 ions. The only difference between the ions is that the azimuth angle increases 0.25 degree between successive ions.

Saving, Loading, and Merging Ion Group Definitions

The top four objects on the Group Definition Screen are used for saving, loading, and merging ion group definition **.FLY** files. *Note:* In addition to the definitions of up to 25 ion groups, the **.FLY** files ***also*** retain the selected coordinate system to use (*from the* **Coordinate System** *selector*).

The Current .FLY File Name

The first object contains the name of the **Current .FLY** file (*if any*). In Figure 8-2, the name is **TEST3D.FLY**.

The Load .FLY Button

The **Load .FLY** button accesses the GUI File Manager to load an ion group definition file. The name of the file loaded will be displayed as the **Current .FLY** file.

The Save Button

The **Save** button accesses the GUI File Manager to save the current collection of ion group definitions as a **.FLY** file. The name of the file saved will be displayed as the **Current .FLY** file.

The Merge Button

The **Merge** button accesses the GUI File Manager to select a **.FLY** file to merge with the current ion group definitions. *Note:* The groups defined in the selected merge file will be ***added to the end*** of any groups currently defined. ***Only*** as many groups will be added as will fit into the ***maximum 25 group limit*** of a **.FLY** file.

Trajectory Group Selection and Group Editing Buttons

A panel object and nine buttons are provided for editing functions involving an ***entire*** ion group.

The Trajectory Group Panel

The **Trajectory Group** panel displays/selects the number of the currently displayed ion group. There are ***two*** numbers on this panel object. The ***leftmost*** is the number of the ***currently selected ion group***. This number can be changed to select the desired ion group to display. The ***rightmost*** number is the total number of ion groups currently defined (*from 1 to 25*).

The Use Electrons Button

The **Use Electrons** button gives ***all*** ions in the ***currently selected ion group*** the rest mass and charge of an ***electron***.

The Use Protons Button

The **Use Protons** button gives ***all*** ions in the ***currently selected ion group*** the rest mass and charge of a ***proton***.

The Default Grp Button

The **Default Grp** button changes the group definition of the ***currently selected ion group*** to a SIMION default group definition.

The Restore Grp Button

The **Restore Grp** button restores the group definition of the ***currently selected ion group*** to what it was when the group was most recently selected via the **Trajectory Group** panel.

The Cut Grp Button

The **Cut Grp** button removes the ***currently selected ion group*** from the ion group definition list and places a copy of it in the ion group paste buffer. *Note: **SIMION will not allow you to cut the last group**.*

The Copy Grp Button

The **Copy Grp** button puts a copy of the ***currently selected ion group*** in the ion group paste buffer.

The Paste Button

The **Paste Grp** button inserts a copy of the ion group paste buffer (*one group definition*) immediately after the ***currently selected ion group***. ***SIMION will refuse to paste a group when 25 groups are already in the group definition list.***

The Swap Up and Swap Dn Button

The **Swap Up** and **Swap Dn** buttons are used to move the ***currently selected ion group*** up (*towards ion group one*) or down (*towards the last ion group*) within the ion group list. ***Ions are flown individually in group order.***

Parameters of Selected Ion Group

Below the ion group editing buttons are the parameters of the ***currently selected ion group***. These parameters can be viewed and edited via their panel objects. The following is a discussion of each parameter.

The Number of Ions Panel

The **Number of Ions** panel is used to specify the number of ions in the ***currently selected ion group***. The number must be one or greater. ***Warning: Ions take memory***. If you request a ***huge*** number of ions SIMION may be forced to virtual (*or it may refuse to fly the ions*).

The Color Panel

The **Color** panel is used to specify the color for ***all ions*** in the ***currently selected ion group***. Any of SIMION's 16 colors (*color index 0-15*) can be selected.

*Note: **Selecting a color of 15 will make the ion <u>invisible</u>**.* This is useful if you are using charge repulsion on a group of ions and only want to see the trajectories of one or more selected ions to keep down the clutter.

If you want ions to have different colors you must define them in different groups.

The CWF Panel

The **CWF** panel is used to specify the ***charge weighting factor*** to use for ***all ions*** in the ***currently selected ion group***. As discussed above, this factor is only used to help in charge repulsion situations. The default value is 1.0.

The Delta TOB Panel

The **Delta TOB** panel is used to set the ***<u>change</u>*** of time of birth between successive ions in the ***currently selected ion group***. The first ion of each group is ***<u>always</u>*** born at time zero. If the ***Delta TOB*** panel value is greater than zero, then each successive ion's birth is delayed by that amount in microseconds.

The First Ion's Mass and Delta Mass Panels

These two panels display and set the ***rest*** masses for ***all ions*** in the ***currently selected ion group*** (*in **unified** atomic mass units*). The first ion in the group has the mass of the **First Ion's Mass** panel. The mass of each successive ion has a **Delta Mass** panel increment applied.

The First Charge and Delta Charge Panels

These two panels display and set the electrostatic charge for ***all ions*** in the ***currently selected ion group*** (*in elementary charges*). The first ion in the group has the charge of the **First Charge** panel. The charge of each successive ion has a **Delta Charge** panel increment applied.

The First X, First Y, First Z, Delta X, Delta Y, and Delta Z Panels

These six panels display and set the starting positions of ***all ions*** in the ***currently selected ion group***. The units and locations depend on the current setting of the ***Coordinate System*** selector (*as described above*). The first ion in the group has the position of **First X**, **First Y**, and **First Z** panels. The position of each successive ion has the **Delta X**, **Delta Y**, and **Delta Z** panel increments applied.

The First Az Angle, First El Angle, Delta Az, and Delta El

These four panels display and set the starting flight direction of ***all ions*** in the ***currently selected ion group***. The units of these angles are in degrees relative to the current setting of the **Coordinate System** selector (*as described above*). The first ion in the group has the orientation of **First Az Angle** and **First El Angle** panels. The orientation of each successive ion has the **Delta Az** and **Delta El** panel increments applied.

The First Ion's KE and Delta KE Panels

These two panels display and set the starting kinetic energy of ***all ions*** in the ***currently selected ion group***. The units of kinetic energy are electron volts (*eV*). The first ion in the group has the kinetic energy of the **First Ion's KE** panel. The kinetic energy of each successive ion has a **Delta KE** panel increment applied.

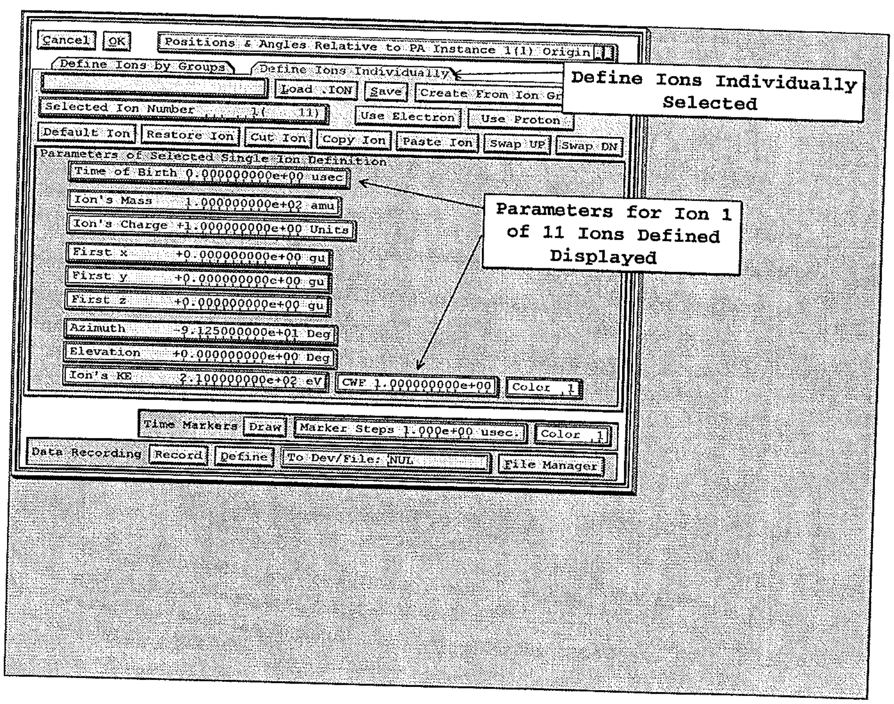


Figure 8-3 Example of the Individual Ion Definition Screen

Defining Ions Individually

Clicking **Define Ions Individually** tab on the Ion Definition Screen accesses the Individual Ion Definition Screen (*Figure 8-3 above*). The Individual Ion Definition Screen allows you to define ions individually. ***This means that each ion's parameters can be totally independent of any other ion's parameters*** (*e.g. color, mass, and etc.*).

The Number of Individual Ion Definitions Allowed

By default, SIMION allows up to ***500*** individually defined ions at one time. If you want to define more than 500 ions you must return to the Main Menu Screen, remove all PAs from RAM, and then adjust the **Allocate Memory for a Maximum of 500 Ions** panel to the desired number of ions. ***Remember not to be too greedy - ions take RAM.***

Saving and Loading Individual Ion Definitions

The three objects at the top left of the Individual Ion Definition Screen are used for saving, loading ion definition **.ION** files. These **.ION** files have an ASCII format to allow you to ***also*** create them outside of SIMION (*see Appendix D*). Moreover, SIMION's data recording feature (*discussed later in this chapter*) can create **.ION** format files too.

Note: SIMION 7.0 (Unlike 6.0) also saves the current coordinate system to use in the .ION file. SIMION 7.0 can read 6.0 **.ION** files but 6.0 can't read 7.0 **.ION** files. See Appendix D for information concerning the differences between the 7.0 and 6.0 **.ION** file formats.

The Current .ION File Name

The first object contains the name of the **Current .ION** file (*if any*).

The Load .ION Button

The **Load .ION** button accesses the GUI File Manager to load an individual ion definition file. The name of the file loaded will be displayed as the **Current .ION** file.

The Save Button

The **Save** button accesses the GUI File Manager to save the current individual ion definitions as a **.ION** file. The name of the file saved will be displayed as the **Current .ION** file.

Converting Ion Group Definitions to Individual Ion Definitions

The **Create From Ion Groups** button is provided to allow you to use the current ion group definitions (***all** currently defined groups*) to create the a list of individual ion definitions. The ions thus created replace any that were in the individual ion definition list. If there are more ions than can be currently defined individually (*500 is the default*), SIMION will warn you and only transfer the number of ions that will fit.

Ion Selection Panel and Editing Buttons

A panel object and nine buttons are provided for editing functions involving an entire ion definition.

The Selected Ion Number Panel

The **Selected Ion Number** panel displays/selects the number of the currently displayed ion. There are ***two*** numbers on this panel object. The ***leftmost*** is the number of the ***currently selected ion***. This number can be changed to select the desired ion to display. The ***rightmost*** number is the total number of individual ions currently defined (*from 1 to 500 or more*).

The Use Electron Button

The **Use Electron** button makes the ***currently selected ion*** have the rest mass and charge of an ***electron***.

The Use Proton Button

The **Use Proton** button makes the ***currently selected ion*** have the rest mass and charge of a ***proton***.

The Default Ion Button

The **Default Ion** button changes the definition of the ***currently selected ion*** to a SIMION default ion definition.

The Restore Ion Button

The **Restore Ion** button restore the definition of the ***currently selected ion*** to what it was when the ion was most recently selected via the **Selected Ion Number** panel.

The Cut Ion Button

The **Cut Ion** button removes the ***currently selected ion*** from the individual ion definition list and places a copy of it in the ion paste buffer. ***Note: SIMION will not allow you to cut the last ion.***

The Copy Ion Button

The **Copy Ion** button puts a copy of the ***currently selected ion*** in the ion paste buffer.

The Paste Button

The **Paste Ion** button inserts a copy of the ion paste buffer (*one ion*) immediately after the ***currently selected ion***. ***SIMION will refuse to paste an ion when the maximum number of ions*** (*typically 500*) ***are already in the individual ion definition list.***

The Swap Up and Swap Dn Button

The **Swap Up** and **Swap Dn** buttons are used to move the ***currently selected ion*** up (*towards ion one*) or down (*towards the last ion*) within the individual ion list. ***Ions are flown individually in ion definition order.***

Parameters of Selected Single Ion Definition

Eleven panel objects are provided to display and edit the definition of the currently selected ion (*via the* **Selected Ion Number** *panel*). The following discusses the use of each of these panels.

The Time of Birth Panel

The **Time of Birth** panel displays the ***delayed*** time of birth for the ***currently selected ion*** in microseconds ***after*** the start of the **Fly'm** clock.

The Ion's Mass Panel

The **Ion's Mass** panel displays the ***rest*** mass of the ***currently selected ion*** in ***unified*** atomic mass units (*amu*).

The Ion's Charge Panel

The **Ion's Charge** panel displays the electrostatic charge of the **currently selected ion** in elementary charge units.

The First X, Y, and Z Panels

The three First X, Y, and Z panels display and set the starting position of the ***currently selected ion***. The units and locations depend on the current setting of the ***Coordinate System*** selector (*as described above*).

The Azimuth and Elevation Panels

These two panels display and set the starting flight direction of the ***currently selected ion***. The units of these angles are in degrees relative to the current setting of the **Coordinate System** selector (*as described above*).

The Ion's KE Panel

The **Ion's KE** panel displays and sets the starting kinetic energy of the ***currently selected ion***. The units of kinetic energy are electron volts (*eV*).

The CWF Panel

The **CWF** panel is used to specify the ***charge weighting factor*** to use for the ***currently selected ion***. As discussed above, this factor is only used to help in charge repulsion situations. The default value is 1.0.

The Color Panel

The **Color** panel is used to specify the color for the ***currently selected ion***. Any of SIMION's 16 colors (*color index 0-15*) can be selected.

Note: Selecting a color of 15 will make the ion <u>invisible</u>. This is useful if you are using charge repulsion on a group of ions and only want to see the trajectories of one or more selected ions to keep down the clutter.

Defining Ions Outside SIMION

You have the option of creating your own **.ION** files outside SIMION. These files are ASCII files (*see Appendix D for their format requirements*). The **.ION** files can be generated either manually (*by you with an editor*) or automatically (*by a program you created*).

Defining Ions Via User Programs

SIMION allows you to ***<u>redefine</u>*** ions within the **Initialize** and **Other_Actions** program segments of user programs. ***See Appendix I for more information***.

This **Initialize** program segment allows you to program SIMION to ***<u>redefine</u>*** the ions for you. The **_RANDOM** demo serves as an example of having user programs randomize your ions for you. The **GROUP.PRG** file in the **_TRAP** demo also provides another example of random ion generation. The best way to make use of this feature is to study how they work and either copy these program segments into your own user program or create one yourself that does just what you want.

The **Other_Actions** program segment allows you to change the mass and/or charge of the ion while it is flying. This is useful if you want to simulate neutralization, fragmentation, or other interesting effects.

Flying Ions

Now that we have defined the ions we want to fly we need to learn how to fly them. SIMION has a collection of ion flying controls on the Normal Controls Screen (*Figure 8-4 below*).

Background Information

The following is provided to help you better understand the ion flying controls and how to effectively use them.

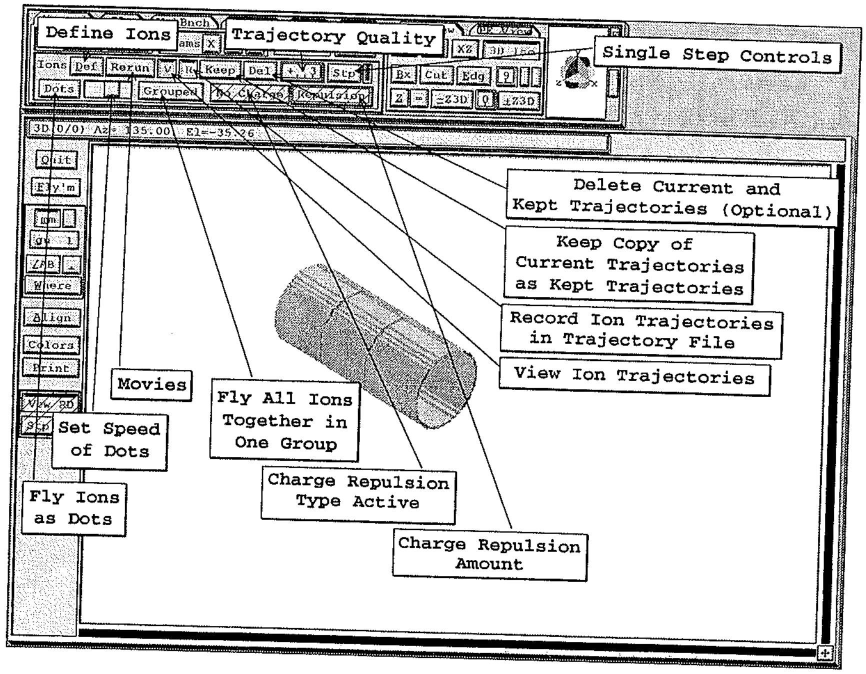


Figure 8-4 Ion flying controls on the Normal Controls Screen

What Can be Done While Ions are Flying

The user interface has been designed to be as interactive as possible. This means that most anything (*e.g. changing views, moving instances, aligning on an instance, and etc.*) can be done ***even*** while ions are flying. This allows you the maximum freedom to change things in a natural way just as if SIMION was a real optics bench that you could fiddle with at any time.

Ions are Flown Individually or in Groups

SIMION allows you to fly each ion defined individually (*one at a time*) or as a group (*all at once*). Generally, flying ions individually is faster. However, flying ions together as a group offers additional options.

SIMION's Clock

SIMION's clock records the ion's time of flight in microseconds. This clock is started (*reset to zero*) each time an ion is flown ***individually*** or when the flying of the entire ion group starts.

If an ion has a delayed time-of-birth ***and*** is flying individually, SIMION will automatically advance the clock to the ion's time-of-birth and start the ion flying.

How Ion Trajectories are Computed

Basically, ion trajectory calculations involve an integration process. SIMION makes use of a highly modified 4th order Runge-Kutta technique (*for details see Appendix E*). In general, SIMION varies the time step used so that an ion moves a fixed amount of distance in a single time step. This distance is normally one grid unit within the current instance.

However, there are times when crossing a boundary, high field curvature, or velocity reversals require smaller time steps to conserve energy properly. SIMION normally checks for velocity reversals and high field curvature (*e.g. grid transitions*) and automatically reduces its time steps via a binary boundary approach method. This strategy provides a robust method for automatically putting the time steps where needed without wasting time steps in simple regions (*e.g. linear fields*).

You have the option of controlling the integration methods used. This allows you to speed calculations in problems that don't require as much protection. Remember, it is ***your*** responsibility to ***verify*** that such compromises are justified (*e.g. the trajectories are not significantly different*).

The Recording of Trajectories

SIMION normally saves the trajectory position data of the ion trajectories in a single ***Fly'm*** to a temporary trajectory plot file. This allows you to view these trajectories ***after*** the end of the flight in different views (*e.g. WB or PE*) and in different ways (*e.g. lines or dots*). ***However, trajectory data can take an enormous amount of disk space***. This is particularly true if your ions are wandering about in an ion trap.

You have the option of turning off (*and erasing the existing trajectory plot file*) the recording of trajectory image data via the **Rerun** or **R** buttons (*see below*).

Viewing Ions as Lines or Dots

Ions are normally displayed as lines. However, it can sometimes be very useful to view them as moving dots (*especially if they are flying in groups*). SIMION allows you to view ions either way via the **Dots** button (*see below*).

Discussion of Individual Ion Flying Controls

The following material gives detailed information about each ion flying control object on the Normal Controls Screen (*Figure 8-4 above*).

The Def Button

The **Def** button allows access to the Ion Definition Screens (*discussed above*). It also allows access to time marker controls as well as SIMION's data recording definition controls.

The Fly'm Button

The **Fly'm** button (***now*** *located on the left edge of the View Screen*) starts the currently defined ions flying when ***depressed***. Clicking the **Fly'm** button again or hitting the **Esc** key ***stops*** the current ion flying process.

Be sure to select the desired flying options (*e.g. Grouped and charge repulsion*) ***before*** starting to fly ions. Selecting this type of option while ions are flying will ***automatically terminate*** the current flight.

If any of the instances have user programs active SIMION will automatically compile them when you depress the **Fly'm** button. If there are any adjustable variables defined in these user programs, SIMION will display a window that allows you to adjust their value(*s*) ***before*** the ions start flying.

The Rerun Button

The **Rerun** button can be clicked during a **Fly'm** and ***will*** ***not*** cause an automatic flight termination. *Note: **The Rerun button can also be controlled by user programs*** (*handy*). This button has three important functions:

1. **First**, it tells SIMION that you want to keep rerunning the trajectories when the **Rerun** button is ***depressed*** at the ***end*** of a **Fly'm**. This feature is handy for movie style animation (*SIMION just keeps flying ions*) that allows you to change voltages or adjust instance positions without having to periodically start the ions flying again.

2. **Second**, when the **Rerun** button is depressed it automatically ***erases*** the current trajectory image file (*if any*) and ***stops*** the recording of trajectory flight image data. A **Rerun** button trick can enhance the power of the **V** and **R** buttons. Let's assume ions are flying, the **Rerun** button is raised, and you want to quickly erase all ion trajectory images that have been saved up to that point (*assuming that the **R** button has been depressed so that ion trajectory images have been saved*). Simply click the **Rerun** button twice (*to lower and **then** raise it*). This trick erases the saved ion trajectories, but doesn't affect either data recording or the re-flying of ions (***Rerun**'s primary function*). This trick can also be accomplished via user programming via the **Rerun_Flym** reserved variable. However, for this trick to work you should set and ***then*** clear the **Rerun_Flym** reserved variable in ***separate*** calls to the other_actions program segment. Remember any reserved variable changes ***only*** take effect when the user program segment actually returns to SIMION (*exits*).

3. **Third**, if data recording to a file is active, SIMION will take note of the status of the **Rerun** button ***at the instant*** when the **Fly'm** button is clicked to ***start*** flying ions. If the **Rerun** button is depressed (*at that instant*) then no recorded data will be sent to the user designated data record file during the ***entire*** **Fly'm**. However, recorded data will always appear on the Data Monitoring Screen (*if and when displayed*).

The V Button

The **V** button controls the viewing of trajectory images (*useable during ion flying*). When it is depressed (*default position*) ion trajectory images are displayed. However, when the **V** button is raised trajectory images will not be displayed. They may still be retained for later viewing. ***Suppressing trajectory drawing can significantly speed up ion flying when data recording is the main interest***. This feature can also be controlled within user programs via the **Trajectory_Image_Control** reserved variable.

The R Button

The **R** button controls the **R**etaining of trajectory images (*useable during ion flying*). When it is depressed (*default position*) ion trajectory images are retained (*Providing that the **Rerun** button is **not** depressed*) for subsequent viewing. However, when the **R** button is raised trajectory images will ***not*** be retained (*despite the status of the **Rerun** button*). This provides a convenient way to record data without having trajectory images saved too! It also keeps trajectory images from filling up your disk (*and speeds up ion flying*) when data recording is the main interest. This feature can also be controlled within user programs via the **Trajectory_Image_Control** reserved variable.

The Keep Button

It is often useful to compare ion trajectories of a previous Fly'm against the current Fly'm. The **Keep** button on the Normal Options Screen has been provided for this purpose. Whenever you have ion trajectories from a Fly'm that you want to keep for comparison, click the **Keep** button. Now you can launch another Fly'm and have both sets of ion trajectories displayed. To delete ion trajectories click the **Del** button *(just to the right of the **Keep** button)*. If trajectories have been kept you will be asked if they should be deleted too.

The Del Button

The **Del** button ***terminates*** any active ion flying and ***erases*** the current trajectory image file (*if any*). Also if the Keep button has been used to create kept ion trajectories, you will be asked if they should be deleted too This is a good way to remove pesky trajectory images from your field of view.

The Trajectory Quality Control Panel

The **Trajectory Quality Control** panel (*showing the value of 3 in Figure 8-4*) is used to control the methods SIMION uses to compute ion trajectories (*see Appendix E*). The default value of 3 turns on ***all*** the computation quality features and provides good trajectory estimates (*at possibly the cost of taking longer*). The value selected for this control panel sets the computational quality in the following manner:

Quality = 0 When the **Trajectory Quality Control** panel is set to zero, SIMION computes ion trajectories at minimum quality (*maximum speed*). Time steps are ***fixed*** for integration steps of one grid unit (*in current instance*) per time step. ***Velocity reversal detection, edge detection, and binary boundary approach are all turned <u>off</u>.*** This level of quality will give totally satisfactory trajectories for many applications.

Quality < 0 When the **Trajectory Quality Control** panel is set to a value less than zero (*e.g.* -2), time steps are ***fixed*** for integration steps of ***1/(1- Quality)*** grid units per time step. Thus a value of -2 would result in time steps of 1/3 grid unit per time step. ***Velocity reversal detection, edge detection, and binary boundary approach are all turned <u>off</u>.***

Quality > 0 When the **Trajectory Quality Control** panel is set to a value greater than zero (*e.g. 3 - the default*), ***Velocity reversal detection, edge detection, and binary boundary approach are all turned <u>on</u>.*** As the quality value increases the edge detection criteria (*CV*) becomes more sensitive. For qualities from 1 to 100 the ***nominal*** time step is set for one grid unit per time step. For qualities above 100 the ***nominal*** time step is set for ***1/(Quality -100)*** grid units per time step.

Fast Ion Flying Between Instances

SIMION 6.0 uses the highest scale value (*mm/grid unit*) of the defined instances for determining the longest time steps when ions were flying outside any instance. Now SIMION 7.0 ***<u>only</u>*** does this when ions are either being flown as dots or some form of charge repulsion is active.

If the ions are being flown as lines ***<u>and</u>*** repulsion is **NOT** active the program uses large time steps to jump between arrays (*and also to and from workbench volume boundaries*). ***This can significantly speed up ion flying if there are long distances between very fine arrays.*** For example, the einzel demo ions fly more than three times faster now.

To force SIMION to not use large time step jumps between arrays, simply ***fly the ions as dots***. Make sure the ion speed is set to maximum (*slider to right edge*), so that there will be no time delay inserted between time steps. ***Note: Clicking the Dots button after the ions have flown (the fly'm is finished) only redraws the trajectories as dots or lines. It doesn't change the time steps of the previously computed ion trajectories.***

The Step and Single Step Buttons

Just to the right of the **Trajectory Quality Control** panel are the **Step** and **Single Step** buttons. Their function is to halt and single step the trajectory calculations.

The Step Button

The **Step** button, when depressed, halts trajectory calculations. When the **Step** button is raised trajectory calculations resume. One use for this button would be to halt a calculation, depress then raise the **Rerun** button to erase any trajectory images, raise the **Step** button for a moment to capture a trajectory section, and then depress it again to halt execution. The trajectory image section then could then be viewed and printed (*if desired*).

The Single Step Button

When the **Step** button is depressed, the button to its immediate right will automatically be raised. This is called the **Single Step** button. Each time you click the **Single Step** button SIMION will execute a ***single*** time step. This is useful for close examination of ion movement from time step to time step.

Another use would be if you are flying ions as dots in a trap and you want to print a picture of the ions at a point in time. Click the **Step** button to halt, depress then raise the **Rerun** button to erase any trajectory images (*or just raise it if it was already depressed*), adjust **Trajectory Quality Control** to -500 (*for the smallest time step*), click the **Single Step** button ***once*** to record a single time step, and then print the desired view.

The Dots Button and the Dots Speed Slider

At the beginning of the next line is a **Dots** button and a **Dots Speed** slider.

The Dots Button

When the **Dots** button is depressed, ions will appear as moving dots during trajectory calculations. This is particularly useful when flying ions in ***groups*** to see how the groups of ions behave (*e.g. in time-of-flight instruments or traps*).

If the **Rerun** button is ***not*** depressed, ions will also appear as dots during display redrawing (*because their images have been recorded*). Depress **Dots** and ions will be redrawn as dots. Click **Dots** again and ions will be redrawn as lines.

If you print trajectories with **Dots** active an ion dot will normally be printed at each time step (*this could be a lot of dots*).

One way to avoid all this clutter is to make use of a time markers trick. Use the **Define** button to access the Ion Definition Screen. Now depress the time marker **Draw** button, set the desired time step (*in microseconds*), and set the marker ***color*** to ***15*** (*the trick*). SIMION will now use each ion's own color for its time markers on any future **Fly'ms**. Further, if **Dots** are active (*button depressed*), ***only*** the time markers (*and **not** the normal dots*) will be sent to the printer (*useful*).

The Dots Speed Slider

The **Dots Speed** slider is used to insert a bit of time delay between time steps when ions are flown as dots. The ***leftmost*** setting inserts a lot of delay (*slowest dots*) while the ***rightmost*** setting inserts no delay (*fastest dots*).

This slider ***is*** accessible during normal trajectory calculations, but ***is not*** accessible when the screen is being redrawn (*e.g. new view or after Fly'm ends*). If you are re-drawing and the dot speed is too slow, hit the **Esc** key to halt drawing, adjust the slider for a faster speed (*move toward right*), move the cursor into the view area, and hit the **Enter** key to force a another view re-draw.

The Grouped Button

The **Grouped** button flies all the ions together in a single group when depressed. This is somewhat (*to significantly*) slower than flying ions individually, because the slowest ion (*the one with the shortest time step*) is the leader of the pack (*or anchor if you prefer*).

This can really slow things down for positive values of **Trajectory Quality**. It is recommended that you always try to use zero or negative values (*fixed steps*) of **Trajectory Quality** to minimize this slow down.

Flying ions in groups is useful to observe their relative movement along the flight line. ***Ions must be flown in groups if charge repulsion is to be used.***

The Charge Repulsion Controls

Access to the charge repulsion controls (*to the right of the* **Grouped** *button*) is blocked unless the **Grouped** button is depressed. These controls are a button that selects the type of charge repulsion used and a panel object that is used to set the amount (*even while ions are flying*).

Grouped ions are normally flown ***without*** charge repulsion (*e.g.* **No Charge** *button displayed*). However, SIMION also supports the ***estimation*** of charge repulsion effects. Note: These computations are useful for determining if you may have charge repulsion problems and where they are occurring. ***It is not intended to model heavily space charged environments accurately***. See Appendix E for more details.

Types of Charge Repulsion Supported

Two basic types of charge repulsion can be modeled (*selected by clicking the* **Repulsion Type** *button and setting the* **Repulsion Amount** *panel*): Ion Cloud and Ion Beam.

Ion Cloud Repulsion

Ion cloud assumes that the defined ions are representative of an actual cloud of ions. Each ion is considered to represent a small cloud of ions (*thus ions are modeled as clouds and not point charges*). The mutual attractions or repulsion between ions (*each mimicking a small ion cloud*) are included in trajectory calculations when ion cloud repulsion is active. The objective is to use a smaller number of grouped ions to simulate a larger group of ions. Two approaches are supported: Coulombic and factor repulsion.

Coulombic repulsion is activated when **Coul Repl** is displayed on the **Repulsion Type** button. SIMION performs coulombic repulsion by dividing a total charge amount in coulombs (*set by user on* **Repulsion Amount** *panel*) among the ions defined (*based on their charge and Charge Weighting Factor*).

Factor repulsion is activated when **Fact Repl** is displayed on the **Repulsion Type** button. SIMION performs factor repulsion by applying a charge multiplication factor (*set by user on* **Repulsion Amount** *panel*) to each ion defined (*Charge Weighting Factor included too*). In this manner one ion can be made to represent 1,000 or more ions in terms of the repulsion effects it has on other ions.

Both coulombic and factor repulsion use ***time coherent integration***. This means all the grouped ions are ***kept together in time***.

Ion Beam Repulsion

The second type of repulsion is beam repulsion. It is selected when **Beam Repl** is displayed on the **Repulsion Type** button. Beam repulsion is modeled by considering ions as lines (*actually - small cylinders*) of charge by using their apportioned current and velocity to compute a charge per unit length. Thus each ion repels or attracts other ions as if it represented a small cloud of line charge (*radius of line is finite*). In order to accomplish this, SIMION apportions the ***ion beam current*** among the group of ions according to their charge and Charge Weighting Factors as above.

Beam repulsion requires a ***special*** type of group flying called ***space coherent integration***. This is due to the fact that ions will not fly at the same speed, but we really need all ions to remain in a plane normal to the axis ion's trajectory so that line repulsion retains some meaning. SIMION accomplishes this by assuming that the ***first*** ion of the ***first*** group (***e.g. first ion defined***) is the axis ion (***at the center of the pack***). ***It is your responsibility to insure this***. It then integrates the trajectory of this ion step by step. At each time step it calculates the normal plane to this ion. All other ions are given the appropriate time steps to keep them roughly on the surface of the plane. ***This means that each ion is really represented at a different time***. However, since this is a beam of ions, one could argue that what we are seeing is the particular ion in the line of ions that currently satisfies the plane intersection requirement.

How to Use Charge Weighting Factors

When a family of ions are initialized you have the option of defining a geometrical weighting factor for each ion group (*or each ion in individual ion definitions*). This is used for apportioning charge around differently when some ions may represent larger initial areas or volumes than others. The formula below shows how CWF effects weighting.

Effective Weight = abs(ion's_charge) * CWF / sum_of_all_ions(abs(ion_charge) * CWF)

How Good is SIMION's Charge Repulsion?

I'm not confident that these calculations are particularly accurate. However, charge repulsion onset levels appear realistic. Moreover, increasing the number of ions used to model the effects doesn't seem to have a dramatic impact on the apparent results (*a small group of ions seems to work pretty well*). ***Beware, this approach doesn't model the space charge effects in ion source regions***.

Flying with Charge Repulsion

The best way to fly with charge repulsion is to use **Rerun** mode. This allows you to keep changing the charge, factor, or current while the ions are flying and have them automatically re-fly with the new settings.

Before clicking **Fly'm**, select **Grouped** and click the **Repulsion Type** button (*to the right*) to select the desired form of charge repulsion: **No Charge** (*repulsion off*), **Beam Repl**, **Coul Repl**, or **Fact Repl**. Now set the current, charge, or factor depending on the charge repulsion mode chosen by using the **Repulsion Amount** panel. ***Click the Fly'm button and they're off***. Note: you may now adjust the amount of repulsion (*current, charge, or factor*) as the ions fly and if **Rerun** is active see the full effects on the next fly cycle. ***Have fun, but don't be too quick to believe what you see***.

Drawing Time Markers

Near the bottom of the Ion Definition Screen (*Figure 8-5 below*) there is a group of controls for generating time markers:

The Draw Button

When the **Draw** button is depressed time markers will be drawn on ion trajectories when ions are flown.

The Marker Steps Panel

The **Marker Steps** panel is used to set the time steps for markers in microseconds.

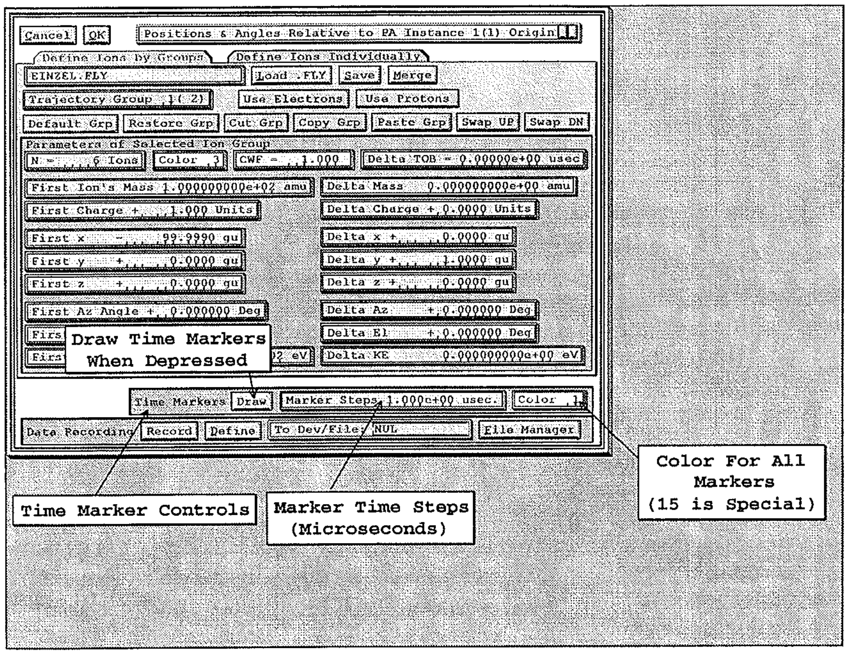


Figure 8-5 Time marker controls on the Ion Definition Screen

The Marker Color Panel

The **Marker Color** panel sets the color of all markers (*including time markers*). Its function is equivalent the that used to set ion colors.

Trick: If you use the color 15 (*White*) for markers, SIMION will actually use the color of the ion for the marker color. Moreover, if you ***print*** trajectories (*having markers drawn with color 15*) with the **Dots** button depressed, SIMION will ***only*** print the ***actual*** markers and not

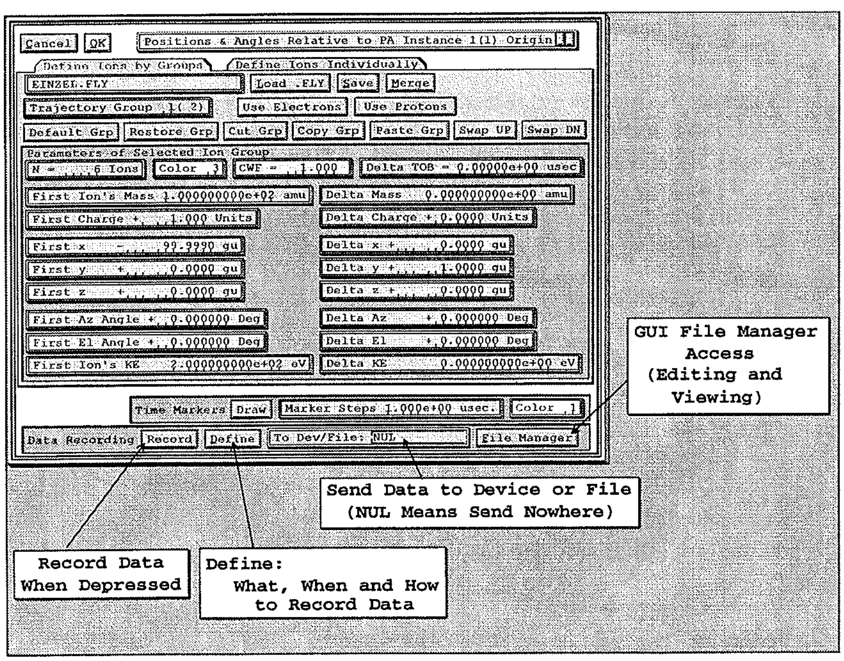


Figure 8-6 Data recording controls on the Ion Definition Screen

the flying dots themselves (*useful*).

Time Markers are Considered Events

Each time marker triggers a data recording marker event. Thus time markers can be used as triggering events for data recording (*discussed below*).

Data Recording

SIMION 6.0 has a powerful data recording option that allows you to record/view specific data, occurring at specified events, in a designated format. Access to these options is via the Ion Definition Screen (*Figure 8-6 above*). At the bottom of this screen are controls for data recording:

The Record Button

If you depress the **Record** button data will be recorded (*and* ***possibly*** *sent to a device or file*) whenever ions are flown. Note: Data is ***never*** sent to a device or file (*but it can be viewed*) when the **Rerun** button is ***depressed prior*** to clicking the **Fly'm** button.

The To Dev/File Ioline

The **Dev/File** ioline object allows you to specify the destination device (*e.g. PRN*) ***or*** file for recorded data. Data is ***always appended*** to the target device/file. ***This means that the recorded data is added to instead of replacing any current data in the target device/file.***

If you output to a device (*e.g.* **PRN**, **LPT1**, **COM1**, *and etc.*) ***do not*** put a colon (**:**) after the device name (*e.g.* **PRN:**).

Note: The default device is **NUL**. This means data is not actually sent to a device/file but still can be viewed on the Data Monitoring Screen.

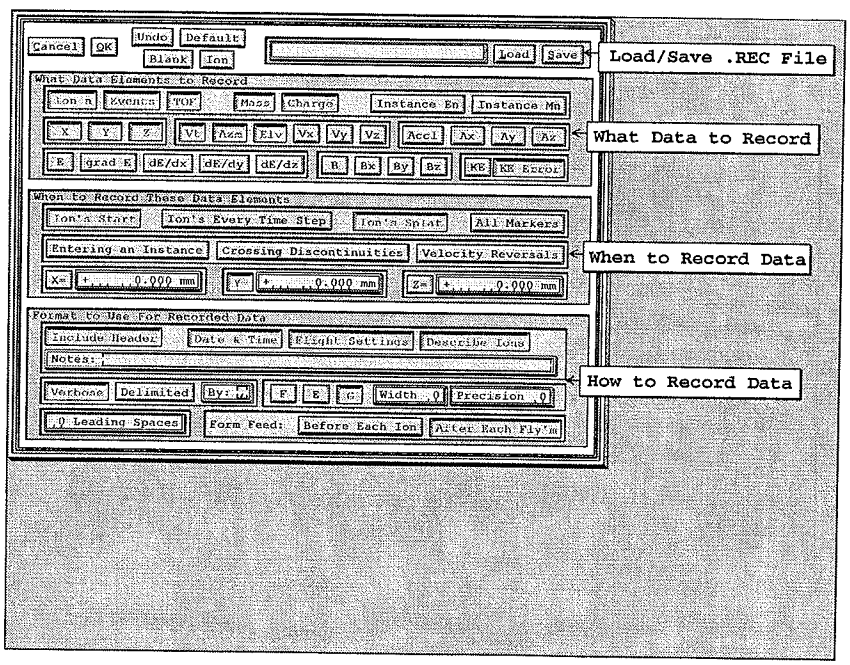


Figure 8-7 Data Recording Definition Screen

The File Manager Button

This button accesses the GUI File Manager. This is useful if you want to inspect the contents of a data recording file with an editor like **EDY** (*supplied with SIMION*). To edit a data recording file click the **File Manager** button, depress the button with the name of the file you want to edit (*with a single mouse button*), and click the **Edit** button on the file manager.

The Define Button

If you click the **Define** button a Data Recording Definition Screen will appear (*Figure 8-7 above*). On it you can define what to record, when to record, and how to record.

The Cancel and OK Buttons

The **Cancel** button exits the Data Recording Definition Screen and ***restores*** the data recording definitions that were active at the time the **Define** button was clicked.

The **OK** button exits the Data Recording Definition Screen and ***keeps*** the current data recording definitions for use in any future **Fly'm**.

Special Editing Buttons

There is a group of special editing buttons at the top center of the Data Recording Definition Screen.

The Undo Button

The **Undo** button restores the data definitions to what they were when the **Define** button was last clicked.

The Defaults Button

The **Defaults** button sets the SIMION default data recording definitions.

The Blank Button

The **Blank** button erases all data recording definitions. This is useful if you want to insure that all options are off ***before*** creating a new set of recording definitions.

The Ion Button

The **Ion** button defines a data recording format that can be saved as an **.ION** file (*it is your responsibility to name it properly - e.g. with an* ***.ION*** *file extentsion*). This is useful if you want to save the termination conditions of your ions for perhaps flying in another workbench (*via the use of hugging and chalking methods*).

Saving and Loading .REC Files

Three objects are provided to allow you to save and load data recording definitions in **.REC** files.

The .REC Filename Ioline

The **.REC Filename** ioline is provided to display the name of the last **.REC** file saved or loaded (*if any*).

The Load Button

The **Load** button is used to load data recording definitions saved in a **.REC** file. The GUI File Manager will automatically assist you in file selection.

The Save Button

The **Save** button is used to save the current data recording definitions in a **.REC** file. The GUI File Manager will automatically assist you.

Selecting What to Record

A collection of buttons are provided to let you select the data elements to record. A data element is selected by ***depressing*** its button. ***The specific help for each button explains the data element and gives its units of measure.***

The Ion n Button

The **Ion n** button requests the recording of the ion's number.

The Events Button

The **Events** button requests that the event (*or events*) that caused the data to be recorded be recorded too. In **Verbose** format a description will be provided. In **Delimited** format a number will be provided that is the sum of the following ***causing*** event(s) related bit flags:

1	Ion Created
2	Next Time Step
4	Hit Electrode
8	Dead in Water
16	Outside Workbench
32	Ion Killed
64	Marker
128	Entering an Instance
256	Field Discontinuity (*CV Limit*)
512	Velocity Reversal
1024	Crossed X = Value Plane
2048	Crossed Y = Value Plane
4096	Crossed Z = Value Plane

The TOF Button

The **TOF** button requests that the time-of-flight (*time since* **Fly'm** *clock started*) in microseconds be recorded.

The Mass Button

The **Mass** button requests that the current ***rest*** mass of the ion in ***unified*** atomic mass units be recorded.

The Charge Button

The **Charge** button requests that the current charge of the ion in elementary charge units be recorded.

The Instance En Button

The **Instance En** button requests that the number of the electrostatic instance (*if any*) that the ion is ***currently*** within be recorded. ***If the ion is not currently in an electrostatic instance a value of zero will be recorded.***

The Instance Mn Button

The **Instance Mn** button requests that the number of the magnetic instance (*if any*) that the ion is ***currently*** within be recorded. ***If the ion is not currently in a magnetic instance a value of zero will be recorded.***

The X, Y, and Z Buttons

The **X**, **Y**, and **Z** buttons request that the current x, y, and z positions in millimeters relative to the ***currently aligned*** ***workbench coordinates*** be recorded.

The Vt Button

The **Vt** button requests that the ion's current speed in millimeters per microsecond be recorded.

The Azm and Elv Buttons

The **Azm** and **Elv** buttons request that the ion's current velocity azimuth and/or elevation orientation angles (*in degrees*) relative to the ***currently aligned*** ***workbench coordinates*** be recorded.

The Vx, Vy, and Vz Buttons

The **Vx**, **Vy**, and **Vz** buttons request that the ion's velocity components (*in millimeters per microsecond*) relative to the ***currently aligned*** ***workbench coordinates*** be recorded.

The Accl Button

The **Accl** button requests that the ion's current total acceleration in millimeters per microsecond squared be recorded.

The Ax, Ay, and Az Buttons

The **Ax**, **Ay**, and **Az** buttons request that the ion's acceleration components (*in millimeters per microsecond squared*) relative to the ***currently aligned*** ***workbench coordinates*** be recorded.

The E Button

The **E** button requests that the electrostatic potential (*in volts*) at the ion's current location be recorded.

The grad E Button

The **grad E** button request that the total electrostatic gradient (*in volts per millimeter*) at the ion's current location be recorded.

The dE/dx, dE/dy, and dE/dz Buttons

The **dE/dx**, **dE/dy**, and **dE/dz** buttons request that the components of electrostatic gradient (*in volts per millimeter*) relative to the ***currently aligned*** ***workbench coordinates*** at the ion's current location be recorded.

The B button

The **B** button requests the total magnetic field (*in gauss*) at the ion's current location be recorded.

The Bx, By, and Bz Button

The **Bx**, **By**, and **Bz** buttons request the magnetic field components (*in gauss*) relative to the ***currently aligned*** ***workbench coordinates*** at the ion's current location be recorded.

The KE and KE Error Buttons

The **KE** button request that the ion's current kinetic energy (*in electron volts*) be recorded.

The **KE Error** button requests that ion's current kinetic energy error (*in electron volts*) be recorded. This error is based on conservation of energy assumptions using the starting and current potential and kinetic energies of the ion. ***Note: This parameter has no meaning if time-changing fields are being applied via user programs.***

Selecting When to Record

The middle collection of buttons designate events that ***will*** trigger a data record when depressed. SIMION will place a dot at the point in the trajectory where the data recording event occurred. The color of the dots will be the color selected for time markers. ***Note: Not all events will be flagged at any computational quality.***

The Ion's Start Button

The **Ion's Start** button designates the start of an ion's flight (*time-of-birth*) as a data recording event when depressed.

The Ion's Every Time Step Button

The **Ion's Every Time Step** button designates ***every*** time step as a data recording event when depressed. This is useful for detailed tracking and troubleshooting. ***Warning: This can generate an enormous number of records.***

The Ion's Splat Button

The **Ion's Splat** button designates the ion's death (*by whatever means*) as a data recording event when depressed. ***Note: Trajectory qualities greater than zero*** (*e.g. the default of 3*) ***are required for good event accuracy.***

The All Markers Button

The **All Markers** button designates all ***time*** markers and ***user program generated*** markers as a data recording events when depressed.

The Entering an Instance Button

The **Entering an Instance** button designates a data recording event when an ion ***just*** enters ***any*** instance (*when depressed*). ***Note: Trajectory Qualities greater than zero*** (*e.g. the default of 2*) ***are required for good event accuracy.***

The Crossing Discontinuities Button

The **Crossing Discontinuities** button designates a data recording event when an ion crosses through a binary boundary (*CV detection based*). ***Note: Trajectory Qualities greater than zero*** (*e.g. the default of 3*) ***are required for this event.***

The Velocity Reversals Button

The **Velocity Reversals** button designates a data recording event when an ion reverses a Vx, Vy, or Vz velocity component in ***currently aligned workbench coordinates. Note: Trajectory Qualities greater than zero*** (*e.g. the default of 2*) ***are required for this event.***

The X, Y, and Z = Plane Crossings

You can designate an event when an ion crosses a specified x, y, or z = value plane. These planes are relative to the ***currently aligned workbench coordinates***. To designate an event plane, click the desired plane's button (**X**, **Y**, *or* **Z**) ***and*** set the desired value (*position in mm*) for the plane in ***currently aligned*** *workbench coordinates.* ***Note: Trajectory Qualities greater than zero*** (*e.g. the default of 3*) ***are required for good event accuracy.***

Selecting the Recording Format

The bottom collection of buttons designate the data recording format. There are two groups of buttons designating: Data Recording Header and Record Format.

Data Recording Header Controls

SIMION can optionally supply a Data Recording Header at the beginning of recorded data for each **Fly'm**. The following objects are used to control its output:

The Include Header Button

The ***Include Header*** button Includes a Data Recording Header at the beginning of data recorded for a single **Fly'm** when ***depressed***. The header will include the information requested by the other header control objects.

The Date & Time Button

The date and time of the start of the **Fly'm** will be recorded when this button is depressed.

The Flight Settings Button

The header will include information concerning grouped flying and repulsion settings when this button is depressed.

The Describe Ions Button

The header will include a description of the ions flown when this button is depressed.

The Notes Ioline

The **Notes** ioline object is included to allow you to include one line of your own personal notes in the header. If the line is blank, no notes will be included in the header.

Data Record Format Controls

The next group of control objects designate the record format used for each data record:

The Verbose and Delimited Buttons

The **Verbose** and **Delimited** buttons are used to select ***either*** verbose ***or*** delimited format data records. The verbose format outputs each data item's name and its value (*designed to be easy for you to read*):

mass(100.0) charge(-1)

The delimited format outputs each data item as a number with a designated delimiter used (*e.g.* ,) to separate each data item. ***A delimited data record is always a single line***. Delimited records are useful for importing data into other programs (*e.g. spreadsheets*).

500,22,15.6

The By: Ioline

The **By:** ioline is used to specify the delimiter to be used to separate data items in delimited format output. The ***comma*** (,) is the default delimiter.

The F, E, and G Buttons

The **F**, **E**, and **G** buttons are used to designate the format to be used for numerical data. ***Note: The <u>user program</u>* Message *cmd makes use of these number format settings <u>too</u>.***

The **F** button creates standard C language f format numerical output using designations for width and precision (*-ddddd.ddddd floating point notation*).

The **E** button creates standard C language e format output using designations for width and precision (*-dddd.ddde-ddd scientific notation*).

The **G** button creates either f or e format output depending on which one is more compact.

The Width and Precision Panels

The **Width** and **Precision** panels are used to set width and precision values that follow C language conventions (*e.g. %width.precision f*). Setting both width and precision to zero gives variable widths under program control (*default*).

The Leading Spaces Panel

You also have the option of inserting a fixed number of leading spaces in front of each line output via the **Leading Spaces** panel.

Form Feed Controls

Buttons are also provided to request form feeds ***before*** each ion flies and ***after*** each **Fly'm**.

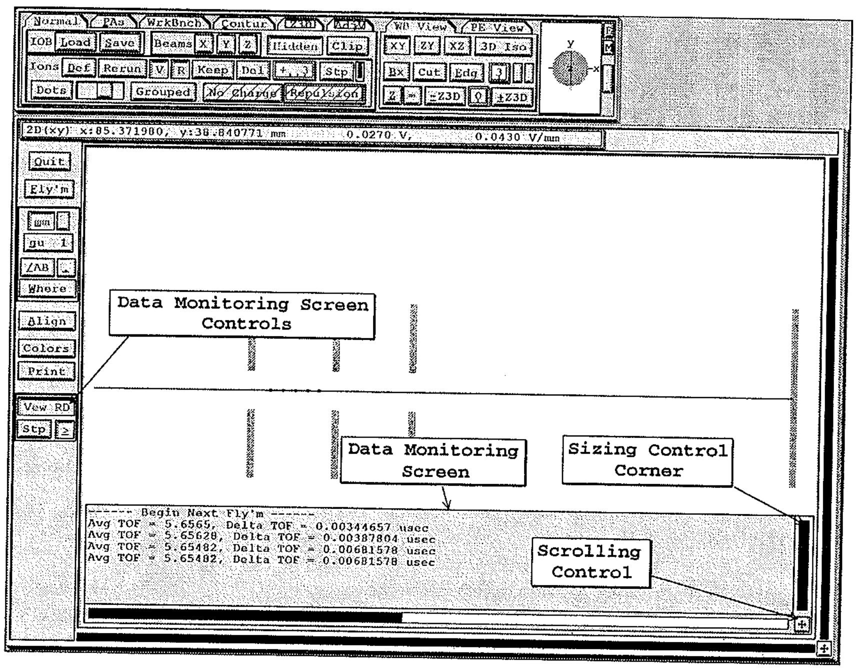


Figure 8-8 View of Data Monitoring Screen in View

Using the Data Monitoring Screen

When data recording is active (*the* **Record** *button is depressed on the Ion Definition Screen) or* when any ***active*** user programs have **Message** or **R/S** commands you will be allowed access to the Data Monitoring Screen controls on the View Screen (*Figure 8-8 above*).

The View Recorded Data Button

The **View RD** button displays the Data Monitoring Screen in the lower left corner of the view window when **depressed**. ***The Data Monitoring Screen is <u>on</u> by default.*** The **View RD** button can be used to turn the screen off and on.

Pausing at Each Recording Event

In order to facilitate data monitoring, a **STP** button has been provided. When this button is ***depressed*** the Ion(*s*) will fly to the next recording event and ***stop*** with the event's data displayed on the Data Monitoring Screen. You can then examine the data at your leisure. To step to the next recording event click the raised '>' button (*to the right of the* **STP** *button*). You may turn **STP** on and off during a flight. ***Note: you may adjust your view, print, or whatever when you are paused.***

Scrolling the Data Monitoring Screen

The Data Monitoring Screen (*visible within the view window when data is being recorded*) has a scroll button (*Figure 8-8*). This button allows you to access the most recent **Fly'm's** last **500** lines of recorded output. ***Access is allowed even while the ions are flying.***

If you scroll with the ***left*** mouse button depressed the data screen will be updated as you scroll. If you scroll with the ***right*** mouse button depressed the data screen will be updated when you ***release*** the mouse button.

The view region bar above the scrolling button is normally ***blue*** indicating that ***all*** lines since the start of the last **Fly'm** can be viewed. The view region bar changes to ***red*** when more than **500** lines have been output to the monitor to warn you that only the ***last*** **500** lines can be viewed (*e.g. records have scrolled off the top of the Data Monitoring Screen*).

Resizing the Data Monitoring Screen

The Data Monitoring Screen can be sized by dragging its upper right corner (*mouse button depressed*). A diagonal arrow will appear to help you find the sweet spot (*Figure 8-8*). The monitor size will be retained ***unless*** you resize the screen (*e.g. by dragging or maximizing*) or exit and reenter View (*then SIMION will restore its preferred data monitor size*).

Flying Ions with User Programs Active

Whenever you click the **Fly'm** button to start ions flying, SIMION quickly checks each instance to see if its referenced potential array has a user program file (*e.g. for* **TEST.PA** *there is a* **TEST.PRG** *file in project directory*). If any potential array has a user program file (**.PRG**), SIMION will ***automatically*** compile its user program segments and use them while flying ions. ***See Appendix I for information on user programs.***

The material presented in this chapter is limited to discussing the special screens you will encounter when flying ions with user programs active. The emphasis will be on how to ***use*** these screens effectively, as ***opposed*** to how to ***write*** user programs.

The Adjustable Variables Screen

If SIMION compiles user programs and finds Adjustable Variables defined therein, it will display an Adjustable Variables Screen just prior to actually flying ions (*Figure 8-9 below*). You can use this screen to change the value of any Adjustable Variable before the ions begin to fly.

Note: Any changes made in Adjustable Variable values will be retained as the proposed value for the next **Fly'm**.

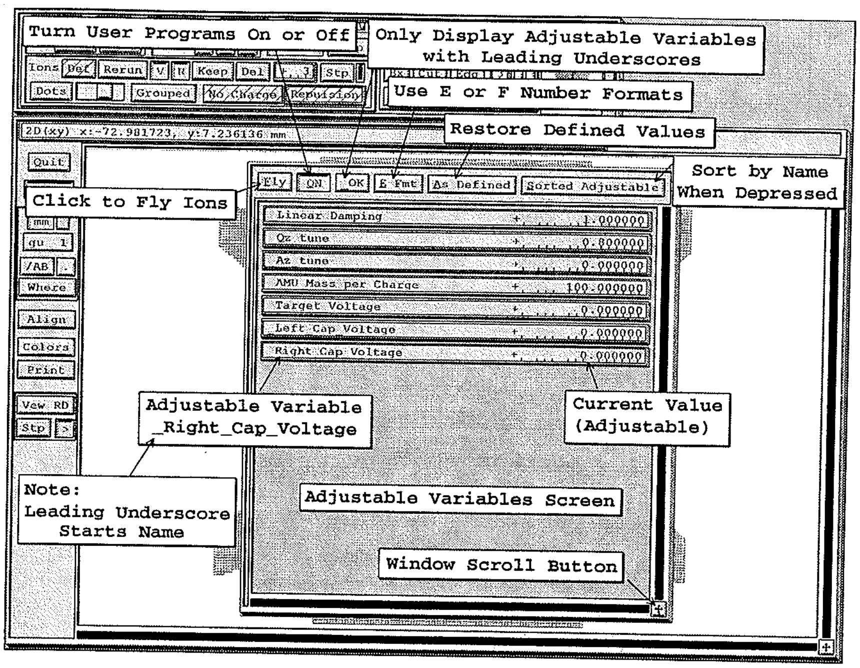


Figure 8-9 Adjustable Variables Screen displayed to support User Programs

The List of Adjustable Variables

Adjustable Variables are displayed as a list of panel objects in a window. The name of the Adjustable variable is on the left side of its panel object, and the value of the Adjustable Variable is on the right side of its panel object.

You can scroll this window vertically via normal GUI methods if more Adjustable Variables are defined than will fit in the window's area.

The Fly Button

The **Fly** button is clicked after you have adjusted the values of the appropriate Adjustable Variables. If you want to abort the **Fly'm**, just hit the **Esc** key.

The ON/OFF Button

The **ON/OFF** button is used to turn ***all*** user programs on or off. The default value is **ON**. This button allows you to turn user programs off for the current **Fly'm**. ***It is useful for testing and debugging purposes***.

Note: You must have Adjustable Variables defined to access this screen and the* ON/OFF *button. Hint: Create a dummy Adjustable Variable to force SIMION to display this screen.

The _OK or ALL Button
Selective Display of Adjustable Variables

It is often desirable to only display an abbreviated list of those Adjustable Variables that the user really would want to change. If ***any*** Adjustable Variable name starts with a leading underscore (e.g. ***_Acceleration_Voltage***) then (*by default*) only it and any other adjustable variables with names beginning with an underscore character will be displayed (*as with the AdjV Screen described below*).

However, it is recognized that there will be times when access to all adjustable variables may be desired. A button has been provided to shift between views of selected (*_OK – Figure 8-9*) and all adjustable variables (*ALL – Figure 8-10 below*). This button ***automatically*** appears at the top of the Adjustable Variable Screen when leading underscore names have been defined for one or more Adjustable Variables. The default is **_OK** for ***only*** displaying leading underscore variables. However, if the button is depressed, the word **ALL** appears and all the adjustable variables will be immediately shown. ***Any changes to the current state of this option is conserved during the remainder of the SIMION session.***

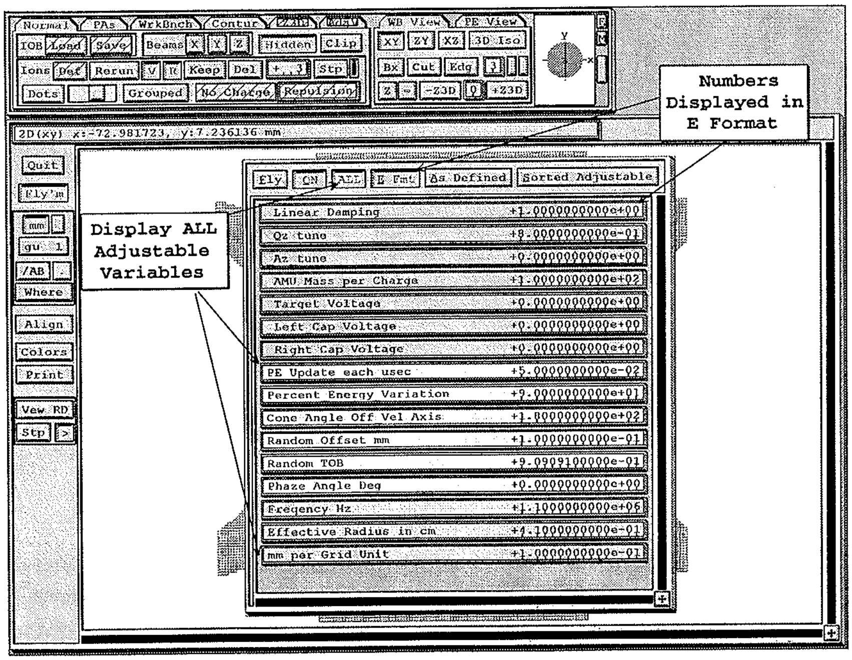


Figure 8-10 Adjustable Variables Screen with ALL variables displayed and E format numbers optioned

The E Fmt Button

The **E Fmt** button displays Adjustable Variable values as ***e*** format numbers ***when depressed*** (*Figure 8-11*) and ***f*** format numbers ***when not depressed*** (*Figure 8-10*). ***Note: F format numbers have a more limited numerical range*** (*be careful*). Your preference will be remembered throughout the current SIMION session.

The As Defined Button

Clicking the **As Defined** button restores the values of ***all*** Adjustable variables to the initial values defined for them in the user programs.

The Sorted Adjustable Variables Button

When this button is ***depressed*** the list of Adjustable Variables will be sorted alphabetically. Otherwise, the list of Adjustable Variables will be in the order they were ***encountered*** during user program compilation.

Accessing Adjustable Variables While Ions Are Flying

There are occasions when it would be very helpful to be able to access Adjustable Variables while ions are flying (*e.g. to change a tune point or damping factor*). SIMION supports this ability via the Adjustable Variables Control Screen accessed via the **AdjV** tab on the View Screen *(Figure 8-11 below)*.

This tab is normally blocked. However, whenever ions are flying with user programs that have Adjustable Variables, the tab is ***unblocked***.

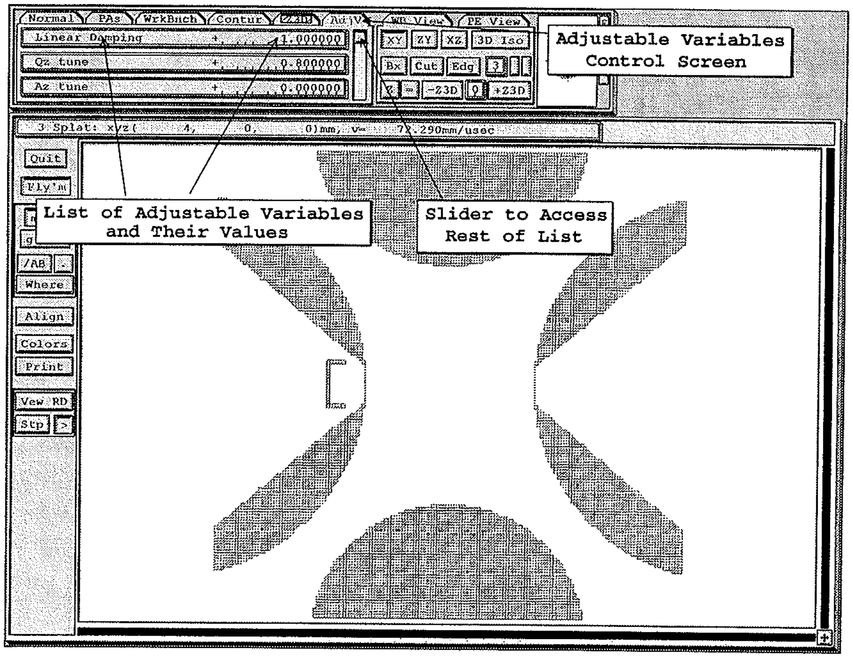


Figure 8-11 The Adjustable Variables Control Screen

The List of Adjustable Variables

The Adjustable Variables Control Screen displays a list of the Adjustable Variables defined. The number display format is either e or f depending on what is currently optioned *(via* **E Fmt** *button in the Adjustable Variables Screen above)*.

The List Access Slider

If there are more than three Adjustable Variables to view, SIMION will automatically provide a ***List Access*** slider on the right edge of the screen.

How the Adjustment Works

SIMION allows you to adjust these Adjustable Variables while the ions are flying. However, it ***does <u>not</u> retain*** any changes you make on this screen. These changes are assumed temporary to the current **Fly'm**.

Moreover, if user programs happen to change the value of any of these Adjustable Variables the displayed value ***<u>will not</u>*** reflect this change.

Thus you should avoid displaying Adjustable Variables that are changed in value by user programs as ions fly. *This can be accomplished by using the trick described below.*

Selecting the Adjustable Variables to Display

If you create names for Adjustable Variables that begin with a leading underscore *(e.g.* **_Linear_Damping**), SIMION will detect this and *<u>only</u>* display Adjustable Variables with leading underscores in the Adjustable Variable Control Screen. Otherwise it will display a list of ***all*** Adjustable Variables defined. Thus if you want to limit the list, ***use leading underscores*** for any Adjustable Variables you want displayed.

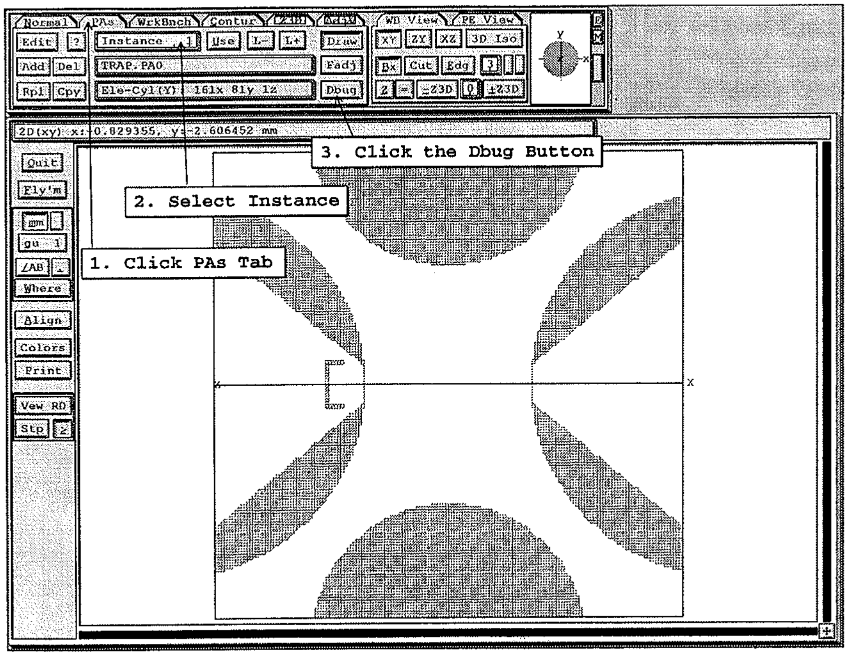


Figure 8-12 Accessing the User Program Development System from within View

Accessing the User Program Debugger From Within View

If you develop user programs you will have bugs and new features you will want to fix or add as quickly as possible. SIMION provides access to the User Program Development System from within the View Screen (*Figure 8-12 above*).

To access the user program development system from within **View**, click the **PAs** tab, use the **Instance Selector** panel to select the instance with the user programs you want to debug, and click the **Dbug** button.