Ion Optics Through the Eyes of SIMION 6.0

AN ASMS SHORT COURSE

Portland ASMS Conference

May 11-12, 1996

Instructors

David A. Dahl
and
Anthony D. Appelhans

Course Outline

Topics - May 11 am	**How**	**Manual**
Get Class Organized	Introduce Instructors	
	State Objectives of Course	
	Explain How Course will be Conducted	
	Divide Class into Working Groups	
Session 1	**Discuss SIMION and GUI philosophy (guided tour)**	**1-1**
	Basic GUI	**F-1**
Learn GUI/SIMION Usage	Objects and Help	**F-6**
	File Manager	**F-12**
	Other GUI Options (Color, Sounds, and etc.)	**F-2**
	Viewing Basics (using EXTRACT.PA)	
	Change View Orientation and Quality	**7-10**
	2D Zoom and Scrolling	**7-12**
	3D Zoom	**7-13**
	Run EINZEL Demo	**C-2**
	Clear Memory and Load with View	**2-12,7-8**
	Load and Fly Ions (alone or grouped)	**8-4**
	Potential Energy Surfaces and Contouring	**7-19,7-34**
	ReRun, Single Step, Fast Adjust with Ions Flying	**8-12,7-28**
	Fly Ions with Charge Repulsion Active	**8-15**
	Printing Views and Screens	**G-1,G-5**
Session 2	**Discuss Radius of Refraction**	**2-4**
	Discuss Ion Definitions	**8-4**
Introduce Ion Optics Concepts	**Defining Ions in Groups (.FLY)**	**8-4**
	Defining Ions Individually (.ION)	**8-8**
	Defining Ions Outside of SIMION (.ION)	**8-10,D-4**
	Discuss Data Recording (.REC)	**8-17**
	Controlling Data Recording to a File	**8-17**
	What, When, How, and to Where	**8-20/25**
	Using the Data Monitoring Screen	**8-25**
	User Programs can Record Data Too!	**I-11**
	Class Lab:	
	Run electrostatic and magnetic refraction demos	
	Compare changes in ion mass	
	Compare changes in ion KE	
	Compare changes in field potentials	
Coffee Break	Get it while you can - during lab	
Session 3	**Discuss Potential Arrays and Refining Methods**	**2-6,6-1**
	Discuss How Potentials are Changed	**4-1,5-1**
Creating and Refining 2D Potential Arrays	**Fast Adjust**	**2-11,6-1**
	Fast Scaling	**2-11**
	Discuss How Magnetic Arrays are Defined and the	**2-10**
	Effect of the ng parameter	**2-10**
	Introduce the Notion of Geometry Files	**J-1**
	Class Lab:	
	Create Fast Adjust Electrostatic Lens	
	Refine Fast Adjust Array	
	Fly Ions and Fast adjust Voltages	
	Create and Refine Fast Scaling Magnetic Lens	
	Fly Ions through Magnetic Lens	
	Use New Function with Canned Geometry File	
Lunch		

Course Outline

Topics - May 11 pm	How	Manual
Session 4 **Time-of-Flight Principles**	**Discuss Use of Instances in Workbench** **Impact of Trajectory Computational Quality** **Discuss Time-of-Flight Concepts** *Class Lab:* Run Linear TOF Demo - with ion bunching Change Computational Quality on Linear TOF Run Reflectron TOF Demo Determine drift time/reflectron time ratios Edit Ions and TOF Instances to change ratio Examine impact of refraction on time focusing	**7-1,7-3** **8-13**
Session 5 **FTMS ICR Cell Principles**	**Discuss the Use of User Programs in ICR Cell Demo** **The Concept of Adjustable Variables** **Discuss ICR Cell Concepts** *Class Lab:* Run ICR Cell Demo View Single Ion Trajectories Pump Kinetic Energy into the Ions View Groups of Ions and Separation by Mass Measure The Cyclotron Frequency with SIMION Trap Ions Formed Outside the ICR Cell	**I-1** **I-14**
Coffee Break	Get it while you can - during lab	
Session 6 **Quadrupole Principles**	**Discuss Use of Instances and User Programs in Demo** **Discuss Quadrupole Concepts** **Quadrupole Entrance and Exit Region Issues** *Class Lab:* Run Quadrupole Demo Examine Trajectory Patterns of Random Ions Find the Three Instance Used for Demo Examine how the Potential arrays are Defined Experiment with Mass Filtering in the Quadrupole Explore How to get Ions Into the Quadrupole	**7-3,I-1**
Session 7 **Ion Trap Principles**	**Discuss Ion Trap Concepts** **Impact of Cooling Gas** **Discuss Use of End-Cap Tickle** **External Introduction of Ions** *Class Lab:* Run Ion Trap Demo on Single Ion Fly Ions Using Damping and Charge Repulsion Tickle Ions with End Caps Inject External Ions Into the Trap Model MSMS Fragmentation in the Trap	

Course Outline

Topics - May 12 am	How	Manual
Session 8 **Creating 3D Arrays With Modify**	**Discuss Modify with 3D Arrays** **2D to 3D and 3D to 2D Conversions** **3D Layers and Marking in Modify** **Find Options** **Replace, Edge, Move, Copy, and etc.** **Discuss the Double and Halve Functions** *Class Lab:* Use Double and Halve and Inspect the Jags	**5-7** **5-9** **5-10,5-14** **5-20** **5-21** **5-4**
Session 9 **Creating Arrays with Geometry Files**	**Discuss Pros and Cons of Geometry Files** **Nested Structure Used in Geometry Files** **Discuss Importance of Locate Command** **Recommend for Geometry File Methods** *Class Lab:* Insert Geometry File in Different Sizes Arrays Modify a Geometry File Create a Functional Geometry File	**J-1** **J-2,J-4** **J-6** **J-21**
Coffee Break	Get it while you can - during lab	
Session 10 **Introduction to User Programming**	**Discuss User Programs and How They are Used** **User Program Segments** **Language Structure** **Adjustable, Static, and Reserved Variables** Class Lab: Connect an Existing User Program File to Array Explore User Program Development System Modify a User Program File Add a New Program Segment Develop Own User Program Code	**I-1** **I-2** **I-4** **I-14**
Lunch		

Course Outline

Topics - May 12 pm	How	Manual
Session 11 **Interest Directed Exploration**	**Reform into Several Interest Groups (as per student interest):** **1. Further Ion Optics Simulations:** Simple Lenses TOFs ICR Cells Quadrupoles Ion Traps or whatever **2. More Advanced SIMION Concepts:** Advanced Instance Tricks Creating a Serious Geometry File Advanced User Programming or some other SIMION topic **3. OR Attacking One or More Problems Suggested by the Instructors.** Students can form the interest groups they desire and transfer between them at will.	
Coffee Break	Get it while you can - during lab	
Interest Directed Exploration Cont.	Second segment gives opportunity for creating new groups, transferring between groups, or continued group exploration as dictated by student interest.	

Session 1

Session 1 - Learn GUI/SIMION Usage

This Lab's Files are found in Directory: C:\CLASS\SESSION1

Learn by doing Discussion (SIMION main menu visible):

1. GUI Objects (buttons, panels, iolines, and windows)
   File Manager(selecting: drive, directory, files, and other options)
   Class Activity (create a subdirectory, copy files into it, and then delete subdirectory)

2. Other GUI Options (colors, sounds, delays, mouse, and video)
   Class Activity (Turn off/on clicks, change colors, change mouse sensitivity, change video resolution)

3. Use Old (load EXTRACT.PA into memory)
   Enter View (demonstrate view orientations and drawing quality options)
   Class Activity (Load EXTRACT.PA and change view orientation and drawing quality)

4. Demonstrate 2D Zoom and Scrolling via area marking
   Demonstrate 2D Zoom and Scrolling via Window button (both page and normal view)
   Class Activity (Practice 2D Zooming and Scrolling)

5. Introduce 3D Zoom Volumes
   Demonstrate 3D Zoom via area marking
   Demonstrate 3D Zoom via the Window Button (page and normal view) and within-view pointing
   Class Activity (Practice the various forms of 3D Zooming)

6. Remove all .PAs from memory
   Load EINZEL.IOB demo into View
   Load Ion .FLY file and fly ions (alone, grouped, as lines, and dots)
   Class Activity (Load EINZEL.IOB, EINZEL.FLY, and fly ions in various ways)

7. Demonstrate Potential Energy Surfaces
   Demonstrate Contouring (2D and 3D)
   Class Activity (Exercise PE surface and contouring options)

8. Demonstrate use of Rerun and how to single time-step ions
   Demonstrate use of Fast Adjust while Ions are Flying
   Class Activity (Exercise Fast Adjust and single time-step options while ions are flying)

9. Discuss/Demonstrate charge repulsion (Factor - beam)
   Class Activity (Practice using beam repulsion with EINZEL)

10. Discuss how to print (view, objects, or entire screens)
    Introduce the various printing options
    Discuss how to use the annotator
    Class Activity (Print a selected view and an entire screen)

This page intentionally left blank

Session 2 - Ion Optics Concepts

Some Basic Ion Optics Concepts

Equations From Basic Physics:

Force = Mass x Acceleration

F = M A

Work = Force x Distance

Forceavg = Work/Distance

F = dW/dr

Coulomb's Law:

$F_e = k\, Q_i\, Q / r^2$ **{Point charge}**

$F_e = Q_i\, Sum_n (k\, Q_n / r_n{}^2)$ **{Point charges}**

Q_i **= The Ion's Charge**

$E = F_e / Q_i = d(W/Q_i) / dr$ **{Electric field intensity}**

E = Volts / Meter

Volts = Joules / Coulomb

$F_e = -e\, E$ **{e = units of positive charge}**

Electrostatic Equation of Motion:

$$\mathbf{A = F / M}$$
$$\mathbf{A = dv / dt = - (e\ E) / m = - E / (m / e)}$$

Magnetic Force Equation:

$$\mathbf{F_m = Q_i\ (U \times B)}\quad \textbf{\{Vector Cross Product\}}$$
$$\mathbf{Q_i} = \textbf{The Ion's Charge}$$
$$\mathbf{U} = \textbf{Vector Velocity of Ion}$$
$$\mathbf{B} = \textbf{Vector Magnetic Field Intensity}$$

Magnetic force is always normal to both the B magnetic field vector and the U velocity component normal to the B magnetic field vector (following the right hand rule):

$$\mathbf{U \times B = (U_yB_z - U_zB_y)i_x + (U_zB_x - U_xB_z)i_y + (U_xB_y - U_yB_x)i_z}$$

A simpler equation results from using ***only*** the vector velocity component normal to the magnetic field:

$$\mathbf{F_m = Q_i\ U_n\ B}\quad \mathbf{\{F_m}\ \textbf{normal to both}\ \mathbf{U_n}\ \textbf{and}\ \mathbf{B\}}$$
$$\mathbf{U_n} = \textbf{Velocity component normal to B}$$

Static Magnetic Equation of Motion:

$$\mathbf{A_n = F / M}$$
$$\mathbf{A_n = (dv/dt)_n = (e\ U_n\ B) / m = U_n\ B / (m / e)}$$

Note: Static magnetic fields change an ion's direction of motion but not its speed (*kinetic energy*).

Refraction in Ion Optics:

Refraction (*bending of ion trajectories*) results from electrostatic and/or magnetic forces normal (*at 90 degrees*) to the ion's velocity.

The Electrostatic Radius of Refraction

Normal Electrostatic Force = Centripetal Acceleration

$$- e E_n = m v^2 / r_n$$

$$r_n = m v^2 / (- e E_n) = -(m/e) v^2 / E_n = -2 KE / (e E_n)$$

$$\text{Where } KE = mv^2/2$$

The Magnetic Radius of Refraction

Normal Magnetic Force = Centripetal Acceleration

$$B_n e v = m v^2 / r_n$$

$$r_n = m v / (B_n e) = (m/e) v / B_n = (2 m KE)^{1/2} / (eB_n)$$

$$\text{Where } KE = mv^2/2$$

Interpretations of Radius of Refraction:

1. The electrostatic radius of refraction is proportional to the ion's kinetic energy per unit charge. Thus all ions with the same starting location, direction, and kinetic energy per unit charge will have identical trajectories in electrostatic (*only*) fields. ***Trajectories are not mass dependent in electrostatic fields***.
2. The magnetic radius of refraction is proportional to the ion's momentum per unit charge. Thus all ions with the same starting location, direction, and momentum per unit charge will have the identical trajectories in static magnetic (*only*) fields. ***Trajectories are mass dependent in static magnetic fields***.
3. Because of the v verses v^2 effects on the radius of refraction, magnetic ion lenses have superior refractive power at high ion velocities.

Light Optics Verses Ion Optics

There are significant differences between light and ion optics:

Radius of Refraction

Light optics make use of sharp transitions of light velocity (*e.g. lens edges*) to refract light. These are very sharp and well defined (*by lens shape*) transitions. The radius of refraction is infinite everywhere (*straight lines*) except at transition boundaries where it approaches zero (*sharp bends*).

Ion optics makes use of electric field intensity and charged particle motion in magnetic fields to refract ion trajectories. This is a distributed effect resulting in gradual changes in the radius of refraction. ***Desired electrostatic/magnetic field shapes are much harder to determine and create since they result from complex interactions of electrode/pole shapes, spacing, potentials and can be modified significantly by space charge***.

Energy (Chromatic) Spreads

Visible light varies in energy by less than a factor of two.

Ions can vary in initial relative energies (*or momentum for magnetic*) by orders of magnitude. ***This is why strong initial accelerations are often applied to ions to reduce the relative energy spread***.

Physical Modeling

Light optics can be modeled using physical optics benches (*interior beam shapes can be seen with smoke, screens, or sensors*).

Ion optics hardware is generally internally ***inaccessible*** and must normally be evaluated via end to end measurements. ***Numerical simulation programs like SIMION allow the user to create a virtual ion optics bench and look inside much like physical light optics benches***.

Session 2

Session 2 - Lab on Radius of Refraction

This Lab's Files are found in Directory: C:\CLASS\SESSION2

Flying ions in a Cylindrical ESA

1. Remove all PAs From RAM
2. Use **View** Function to load **LOWELECT.IOB** (cylindrical ESA)
   Click **YES** to restore potentials.
3. Click the **Define** button (to access Ion Definition Screen) and load **2LOWKE.FLY** ion family file (two 100 eV ions: 100 amu - red, 1,000 amu blue).
4. Click the **Define** button (bottom of Ion Definition Screen) to access Data Recording Screen.
5. Load the **RECORD.REC** data recording definition file.
6. Click **OK** to return to the Ion Definition Screen.
7. Depress the **Record** button to turn data recording on.
8. Click **OK** to return to the View Screen.
9. Click **Fly'm** button to fly the ions. Write down the time-of-flight (TOF).
10. Fly the ions grouped as dots and watch.

Notice that both ions have the same trajectory, although their velocities - time-of-flight TOF is different (*ion trajectories in electrostatic fields are KE not mass dependent*). Verify that the Changes in TOF relate to changes in ion mass.

Changing Ion KE and electrostatic field strength

11. Click **PAs** tab and then **Fast Adjust** button and ***write down*** the Electrostatic Potential
12. Change Potential (play a bit), re-fly the ions and look at the results (ReRun can be used for movies).
13. Change Ion's Energies, re-fly the ions and look at the results.

Compensating for Increasing Ion KE within an electrostatic field

14. Use **Load** button within **View** Screen (***not*** from Main Menu Screen) to load **HIELECT.IOB** file. Click **YES** to restore potentials (required for higher energy ions).
15. We have increased the electrostatic field (via HIELECT.IOB). Now we need to load and fly higher energy ions. Click the **Define** button (to access Ion Definition Screen) and load **2HIGHKE.FLY** ion definition file (two 1,000 eV ions: 100 amu - red, 1,000 amu blue).
16. Click **OK** to return to the View Screen.
17. Click **Fly'm** button to fly the ions.

Notice that the ions' trajectories are unchanged (although TOF is shorter). The higher electrostatic potential just compensates for the ions' higher KE. Verify that the **change** in TOF is appropriate for the change in KE.

18. Click **PAs** tab and then **Fast Adjust** button and ***write down*** the new higher electrostatic Potential

Compare the electrostatic potential needed for the 1,000 eV ions to that needed for the 100 eV ions. What is the factor? Is it logical? Why?

More on next page!

Flying ions in a Magnetic Field

1. Use **View** Function on Main Menu Screen to load **HIGHMAG.IOB** cylindrical ESA (after removing all PAs from RAM).
   Click **YES** to restore magnetic potentials (required for lower energy ions).
2. Click **Fly'm** button to fly the ions.

Notice that both ions have the **different** trajectories (ion trajectories in magnetic fields ***are*** **mass** dependent). Write down the TOF.

Changing Ion KE and magnetic field strength

3. Click **PAs** tab and then **Fast Adjust** button and write down the Magnetic Potential
4. Change Potential (play around), re-fly the ions and look at the results (ReRun can be used for movies).
5. Change Ion's Energies, re-fly the ions and look at the results.

Compensating for decreasing Ion KE within a magnetic field

6. Use **Load** button within **View** Screen (***not*** from Main Menu Screen) to load **LOWMAG.IOB** file (simple two pole magnet).
   Click **YES** to restore magnetic potentials (required for lower energy ions).
7. We have decreased the magnitic field (via LOWMAG.IOB). Now we need to load and fly lower energy ions. Click the **Define** button (to access Ion Definition Screen) and load **2LOWKE.FLY** ion definition file (two 1000 eV ions: 100 amu - red, 1,000 amu blue).
8. Click **OK** to return to the View Screen.
9. Click **Fly'm** button to fly the ions.

Notice that the ions' trajectories are unchanged (although TOF is longer). The lower magnetic potential just compensates for the ions' lower KE. Verify that the **change** in TOF is appropriate for the change in KE.

10. Click **PAs** tab and then **Fast Adjust** button and write down the new lower Magnetic Potential

Compare the magnetic potential needed for the 1,000 eV ions to that needed for the 100 eV ions. What is the factor? Is it logical? Why?

More for the Swift

1. There is a tiny beam-stop instance used in these demos can you find it and determine its shape and orientation?

2. Change the data recording so that data is also recorded at a Splat. Or record some other data. Try turning the data monitoring screen display on and off (from View Screen). Stop ions at each data recording point and then step to the next recording point.

4. Practice your viewing skills. Turn the drawing of electrostatic and magnetic arrays on and off. Cut each demo up and look inside. Try PE views (remember PE views have little meaning with magnetic arrays).

5. Verify that the magnetic radius of curvature is accurate: $r_{mm} = 1440\ (m_{amu}V_{eV})^{1/2}/B_{gauss}$
   Verify that the electrostatic radius of curvature is accurate: $r_{mm} = 2\ V_{eV}/Egradient_{V/mm}$

Session 3 - Creating and Refining 2D Arrays

Determining Field Potentials

The electrostatic or magnetic field potential at any point within an electrostatic or static magnetic lens can be found solving the Laplace equation with the electrodes (*or poles*) acting as boundary conditions.

The Laplace Equation

$$\mathbf{DEL^{2}\ V = 0}$$

The Laplace equation constrains all electrostatic and static magnetic potential fields to conform to a zero charge volume density assumption (*no space charge*). ***This is the equation that SIMION uses for computing electrostatic and static magnetic potential fields***.

Poisson's Equation Allows Space Charge

$$\mathbf{DEL^{2}\ V = - p / e}$$

Poisson's equation allows a non-zero charge volume density (*space charge*). ***SIMION does not support Poisson solutions to field equations. It does however employ charge repulsion methods that can estimate certain types of space charge and particle repulsion effects***.

The Nature of Solutions to the Laplace Equation

The Laplace equation defines the potential of any point in space in terms of the potentials of surrounding points.

For example, Laplace equation is satisfied (*to a good approximation in 2D*) when the potential of any point is ***estimated*** as the average of its four nearest neighbor points:

$$\mathbf{V = (V_1 + V_2 + V_3 + V_4) / 4}$$

The Potential Array

SIMION utilizes potential arrays to define electrostatic and magnetic fields.

A potential array can be either electrostatic or magnetic <u>but not both</u>. If you require both electrostatic ***and*** magnetic fields in the same volume, instances of electrostatic ***and*** magnetic arrays must be ***superimposed*** in the workbench.

A potential array is an array of points organized into equally-spaced square (*2D*) or cubic (*3D*) grids.

All points have a potential (*e.g. voltage*) and a type (*e.g. electrode or non-electrode*). ***Groups of electrode points create the electrode and pole shapes (<u>the finer the grid the smoother the shapes</u>)***.

How Potential Arrays Are Created and Defined

Potential arrays are normally created using the New function and defined by the **Modify** function or Geometry Files.

Potential Array Dimensions

Potential arrays are dimensioned by the number of points in each dimension (*x, y, and z*).

All arrays have their lower left corner at the origin (*xmin = ymin = zmin = 0*). Thus if nx equals 51, xmin equals 0 and xmax equals 50 (*a **width** in **x** of **50** grid units*).

All 2D arrays have nz = 1 (*zmin = zmax = 0*). Thus 2D arrays are always located on the z = 0 xy-plane.

***Arrays can use a lot of memory*. *10 megabytes are required for each million array points*.**

Potential Array Symmetries

Two potential array symmetries are supported: ***Planar*** (*2D and 3D arrays*) and ***cylindrical*** (*2D arrays only*).

Note: The cylindrical 2D array is visualized as if it were 3D planar to take advantage of fast planar visualization methods.

When SIMION projects a 2D planar array as an instance within the workbench volume it assumes a z depth of +/- ny (*grid units*). ***You may edit the instance definition to increase this z depth*.**

The Use of Array Mirroring

***Designs that mirror in y have a mirror image of the electrode/poles in the negative y*.** SIMION supports three mirroring symmetries: x, y, and z.

Mirroring allows you to use a ***smaller*** array to model a larger area (*or volume*) when conditions permit. ***SIMION takes mirroring into account when refining arrays and projecting their instances into the workbench volume*.**

X mirroring is ***allowed*** for all 2D and 3D arrays.
Y mirroring is ***required*** of 2D cylindrical and ***allowed*** in 2D and 3D planar arrays.
***Z mirroring is only allowed in 3D planar arrays*.**

Potentials and Gradients in Potential Arrays

SIMION uses the same methods for refining both electrostatic and magnetic potential arrays. **However, the potentials and gradients used are different and it is important that you understand these differences**.

Potentials and Gradients in Electrostatic Arrays

Potentials in electrostatic arrays are ***always*** in volts.
Electrostatic field gradients are ***always*** in volts/mm.

When an instance of a potential array is projected into the workbench volume it is scaled by a user specified number of millimeters per grid unit (*or defaulted to 1 mm/grid unit*):

Gradient = PA Gradient / Instance Scaling

Volts / mm = (Volts / grid unit) / (mm / grid unit)

Potentials and Gradients in Magnetic Arrays

SIMION is not a magnetic circuit program. You have to supply the magnetic potentials for it to refine. Unlike electrostatics, we normally think of ***and measure*** flux (*gauss*) as opposed to potentials.

In order to deal with this dilemma SIMION defines magnetic potentials in Mags.

Mags are defined to be ***gauss times grid units.***

Thus instance scaling has no impact on the magnetic fields (flux in gauss) produced by magnetic potential arrays.

The ng Magnetic Scaling Factor

The* ng *scaling factor has been provided to further simplify your life. It allows Mags to be directly related to gauss.

Let's say we have a simple two pole magnet. We would like to set one pole to 1000 Mags and the other to Zero Mags and have the field in between be approximately 1000 gauss. If the two poles are separated by let's say a 60 grid unit pole gap we would specify the value of 60 for **ng** to automatically scale the Mags potentials ***roughly*** into gauss.

PA Magnetic Flux = PA Magnetic Gradient (PA gauss)
Magnetic Flux = PA Magnetic Flux * ng

gauss = PA gauss * ng (pole gap scaling factor)

Note: The B field vector ***always*** points from greater magnetic potential (*e.g. 1000 Mags*) toward lesser magnetic potential (*e.g. 0 Mags*).

Beware! Magnetic poles do not as a rule have totally uniform magnetic potentials across their surfaces (*permeability not being infinite*). Thus you must allow for this fact if the effects could be significant enough to impact your results.

Adjusting Array Potentials

Modify the Potential(s) and Re-Refine

The first strategy is the brute force approach:

1. Use the **Modify** function to change the potentials of the points of one or more electrodes or poles in the potential array.
2. The use the **Refine** function to re-refine the potential array to obtain the resulting potentials of the non-electrode or non-pole points.

This is the **Modify-Refine** cycle of array potential adjustment. It is time consuming and invites errors (*e.g. accidentally not changing the potentials of **all** the points of an electrode or pole*).

Proportional Re-scaling of All Array Potentials (.PA arrays)

SIMION 6.0 supports proportional re-scaling of all **.PA** array potentials. This is useful in those cases when the potentials of all electrodes or poles can be changed proportionally to obtain the desired result. ***This approach works with any simple array (with .PA file extension) that has already been refined:***

1. Use the **Fast Adjust** function to change the potential of the highest potential electrode or pole in the potential array.
2. SIMION will automatically scale the potentials of all array points (*electrode/pole and non*) by the same proportion that you changed the potential of the highest potential electrode or pole.

Using Fast Adjust Arrays (.PA# arrays)

The third approach supported by SIMION makes use of the additive solution property of the Laplace equation. ***Fortunately, SIMION is designed to do all the hard work for you!***

1. Use the **Modify** function to define the geometry of your electrodes. ***All points of the first adjustable electrode/pole must be set to exactly 1 volt/Mag***. Up to 30 adjustable electrodes can be defined in this manner within a potential array.
2. Save this potential array with a **.PA#** file extension to signal to SIMION that this is a Fast Adjust Definition File (*e.g. save as* **TEST.PA#**).
3. Refine the **.PA#** file. SIMION will ***recognize*** that this is a Fast Adjust Definition File and create, refine, and save each required solution array automatically (**.PA0** and so on). The **.PA0** array is the base solution array (e.g. **TEST.PA0**).
4. Use the **Fast Adjust** function on the **.PA0** file to set potentials . ***If you want to save the current potential settings of a* .PA0 *file between SIMION sessions, simply save the* .PA0 *file to disk***.

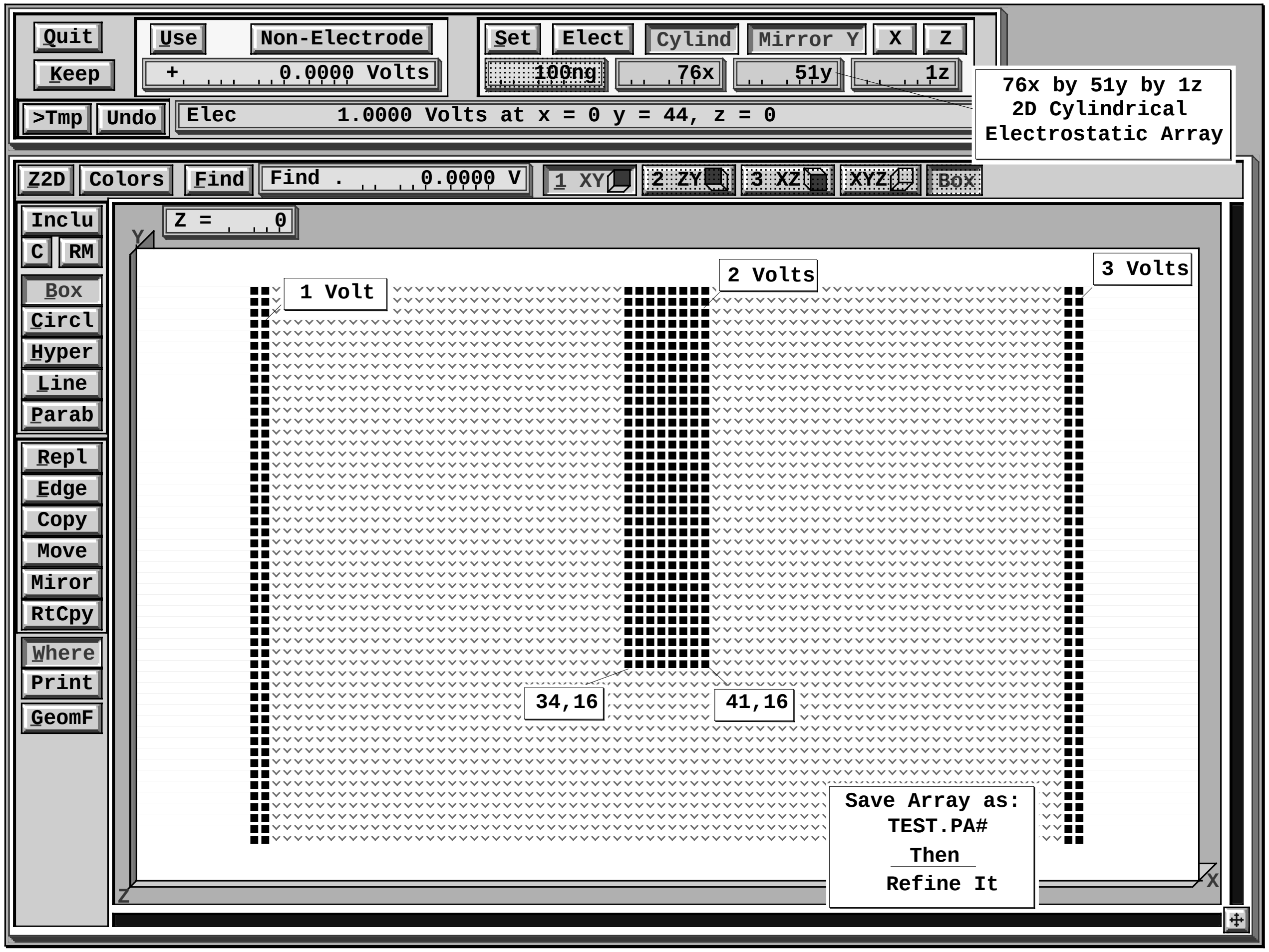
Session 3

<u>Session 3 - Lab on 2D Potential arrays - No After Installation Files Need to be Created</u>

<u>This Lab's Files are found in Directory: C:\CLASS\SESSION3</u>

<u>Creating and Refining a Simple Fast Adjust Electrostatic Array</u>

1. Remove all PAs From RAM
2. Use GUI File Manager to insert a directory called PROJECT below the \CLASS\SESSION3 directory and switch to this new directory.
   a. Click the GUI File Manager button (Main Menu Screen)
   b. Click on the \CLASS\SESSION3 directory (to switch to it)
   c. Click the Other button (to access other options)
   d. Enter PROJECT in the Create Directory ioline and press ENTER.
   e. Click on the \CLASS\SESSION3\PROJECT directory (to switch to it)
   f. Click the Use button to exit the GUI File Manager and stay in the currently selected directory.
3. Click the New button and create a 76x by 51y by 1z 2D cylindrical electrostatic array.
4. Click the Modify button and use Modify to define the electrodes as show in the illustration below. Note: If you have made an error in defining the array in Create you can change the definition from within Modify (using the SET button - after making the desired definition changes).
5. When you're satisfied that the results are OK, exit Modify by clicking the Keep button.
6. The next task is to save the array as a fast adjust array (.PA#).
   a. Click the Save button, and the GUI File Manager will appear.
   b. Make sure you are in the proper directory: \CLASS\SESSION3\PROJECT (click the proper directory if needed).
   c. Enter the name TEST.PA# in the Filename ioline and Press ENTER.
   d. A file memo entry screen will appear. Either enter a short memo or press ENTER to skip the memo.
7. You will now be back on the Main Menu Screen. Click the REFINE button to enter the Refine function.
8. Click the Refine button on the Refine Screen and SIMION will create and refine all the required fast adjust potential arrays.
9. Now click the Fast Adjust button (on the Main Menu Screen). SIMION will automatically load the TEST.PA0 file (your base solution file). Note: You could have loaded this file first via the Old or Load buttons.
10. Set the left and center electrodes to 1,000 volts and the right electrode to 0 volts. Now click the Fast Adjust button to change the potentials in the TEST.PA0 array.
11. You will now be back on the Main Menu Screen. Click the View button to enter the View function. SIMION will automatically create a workbench volume the size of the potential array (TEST.PA0) assuming one mm per grid unit scaling.
12. Use various view orientations to look at the array including potential energy views.
13. Click the Define button on the Normal Options Screen to access ion definitions. Create a group of 11 positively charged ions (blue) that start at x = 1 y = -5 (dy of + 1) with an energy of 0.01 eV.
14. Click the OK button to return to View and fly the ions. They should focus. This is a converging acceleration lens.
15. Click the Define button on the Normal Options Screen to access ion definitions again.
16. Click the Copy Grp button and then the Paste Grp button to create a copy of the current ion group.
17. Change the color of the second ion group to red, its charge to minus one, and the starting energy to 1050 eV.
18. Click the OK button and re-fly the ions. Use the PE view to see the negative ion surface. The positive ions see this lens as a converging acceleration lens, while the negative ions see this lens as a diverging deceleration lens.
19. Now click the PAs tab to access the PAs Option Screen. Click the Fast Adjust button and change the center electrode potential to zero volts. Re-fly the ions view the results. You now have a diverging acceleration for positive ions and a converging acceleration for negative ions.



```
Elec          1.0000 Volts at x = 0 y = 44, z = 0
```

76x by 51y by 1z
2D Cylindrical
Electrostatic Array

1 Volt

2 Volts

3 Volts

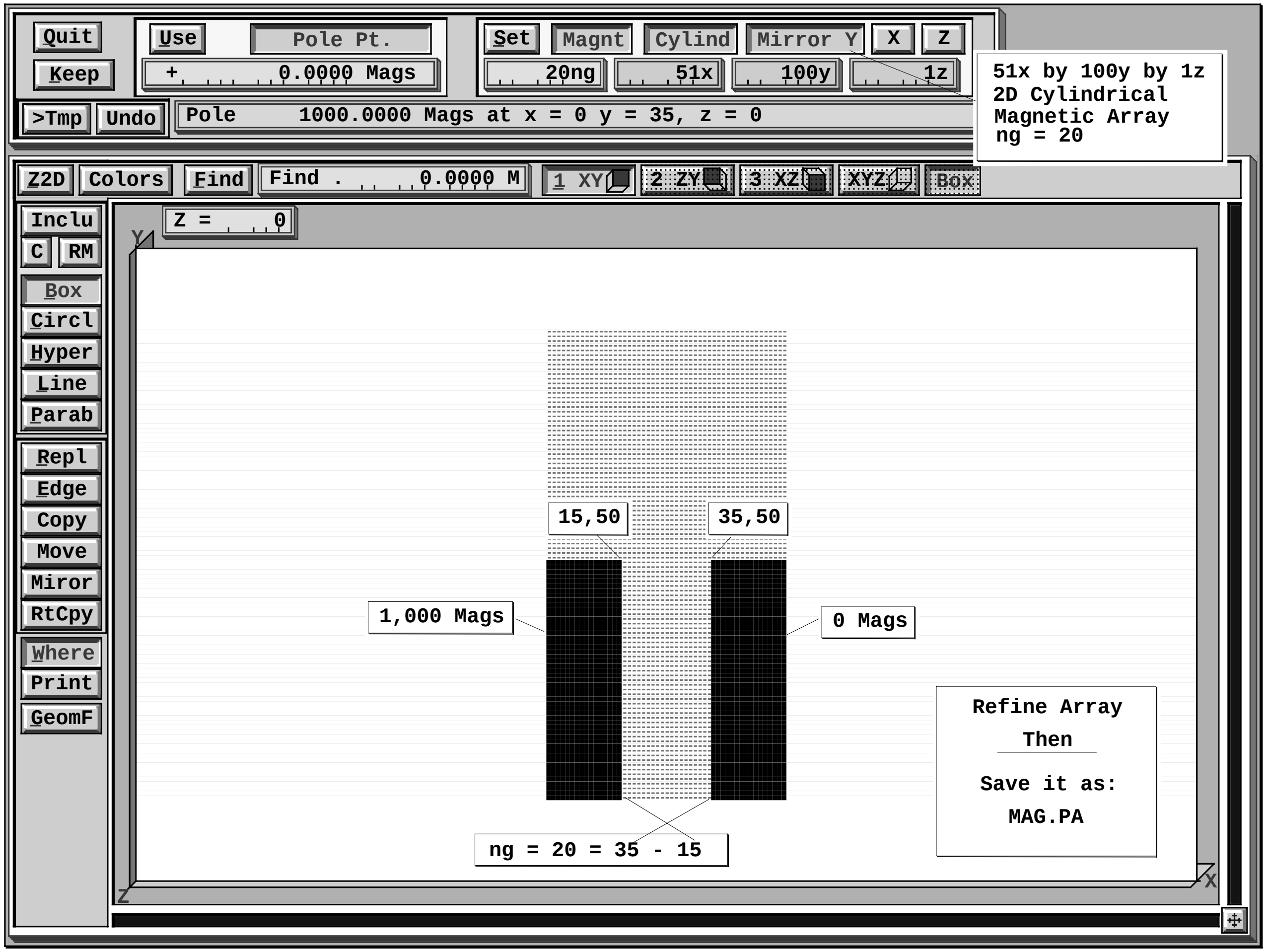
34,16

41,16

Save Array as:
TEST.PA#
Then
Refine It

Creating and Refining a Magnetic Potential Array

1. Remove all PAs From RAM
2. Click the New button and create a 51x by 100y by 1z 2D cylindrical magnetic array with ng = 20 (we plan to have a 20 grid unit pole gap).
3. Click the Modify button and use Modify to define the poles as show in the illustration below. Note: If you have made an error in defining the array in New you can change the definition from within Modify (using the SET button - after making the desired definition changes).
4. When you're satisfied that the results are OK, exit Modify by clicking the Keep button.
5. Now use Refine to refine the array (remember this is a normal PA not a fast adjust definition array).
6. Since this is not a Fast Adjust array we must save the refining results to keep them between sessions. Click the Save button and save the magnetic array as MAG.PA.
7. Enter View and look at the array.
8. One pole is 1,000 Mags, the other is 0 Mags, the pole separation is 20 gird units, and the value for ng is 20. Move the cursor to the center of the magnet and verify that the field intensity is 1,000 gauss. Note that the field intensity decreases quickly as you exit area between the poles.
9. Now try to define a group of 1,000 eV ions that pass between the poles and are bent by the magnetic field. This may take more effort than you expect because of the orientation of the potential array (the x, y, and z workbench directions align with the array by default).
10. Once you get ions flying through the magnet use the Fast Adjust button on the PAs Option Screen to fast scale the field to some other field intensity and re-fly the ions.
11. You might also want to try expanding the workbench volume by clicking on the WrkBnch Tab and adjusting its size - be brave.



```
Pole      1000.0000 Mags at x = 0 y = 35, z = 0
```

51x by 100y by 1z
2D Cylindrical
Magnetic Array
ng = 20

15,50

35,50

1,000 Mags

0 Mags

ng = 20 = 35 - 15

Refine Array
Then
Save it as:
MAG.PA

Session 3

Demo of Using a Geometry File to Create an Array

1. Remove all PAs From RAM
2. Click the New button.
3. Click the Use Geometry File button and the GUI File Manager will appear.
4. Switch to the \CLASS\SESSION3 directory.
5. Click ***both*** mouse buttons on the MAGNET.GEM button to select and use it. SIMION will load and use the definitions in this .GEM file to create the same magnetic array you created in the example above.
6. Click the Modify button to verify that the array and its pole definitions were defined properly.
7. Let's now take a quick look at a geometry file:
   a. Click the GeomF button on the Modify Screen to access the Geometry file development system.
   b. Click the Edit button to load the MAGNET.GEM file into the EDY editor.
   c. Look at the file. These commands created and defined the potential array. More of this tomorrow.
   d. Hit ESC then Q then N to exit the EDY editor.
   e. Click the Quit button to return to the Modify Screen.
8. Click the Keep button to keep the array and exit Modify.
9. Refine the array (it is a simple .PA array).
10. Save the array as MAGNET.PA (needed to save solutions for use by .IOB file).
11. Remove all PAs from RAM.
12. Click the View button and load MAGNET.IOB and restore potentials.
13. Click the Define button (within View) and load the MAGNET.FLY ion definition file.
14. Click OK to return to View and fly the ions.
15. This is an example of what is known as Z focusing. The entry points of the ions control whether they are focused or defocused in the z direction.

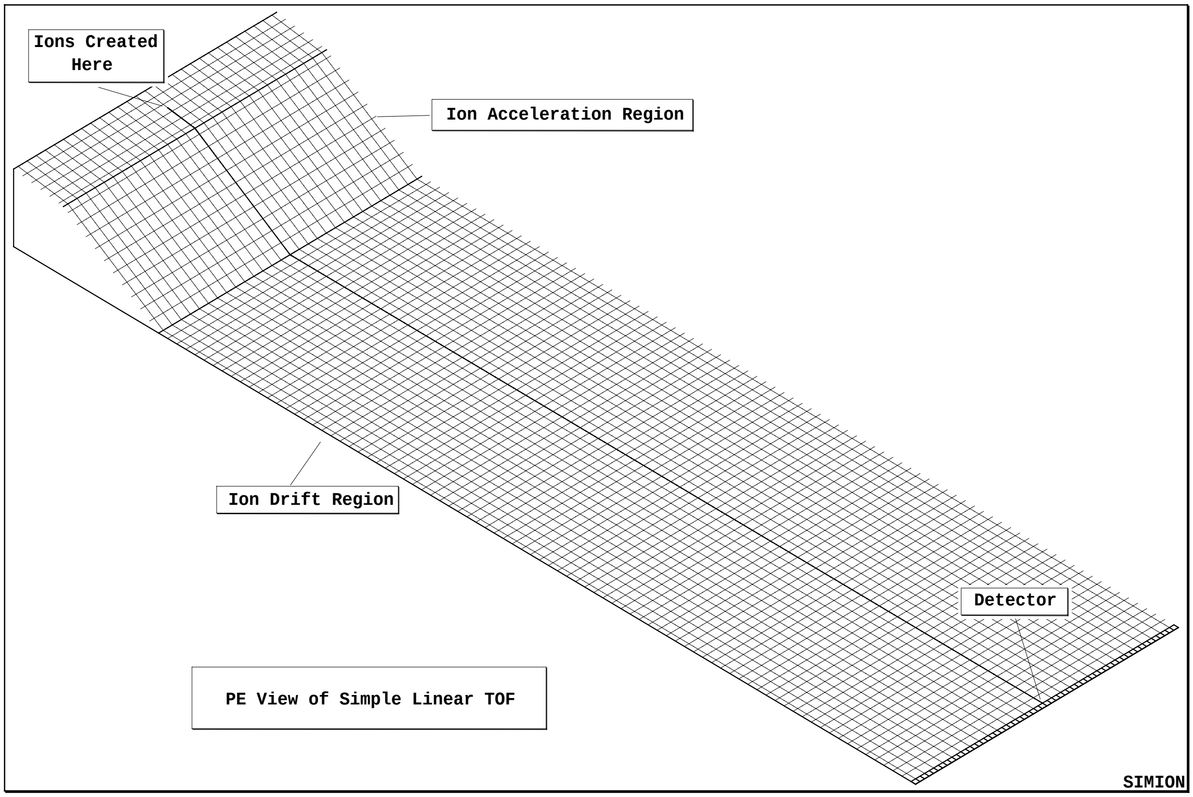
More for the Swift

Note that the orientation of the magnetic array is better in the demo than what you had. This is because the instance definition specified a different orientation. To examine it:

1. Click the PAs Tab.
2. Click the Edit button. You are now looking at the first instance editing screen. Things like scaling, offsets and integral orientations appear on this screen.
3. Click the More button to see the second instance edit screen. Note that orientation of the array has been swung 90 degrees in the azimuth direction. Moreover the working origin of the potential array has been shifted a +25 grid units so that it is in the center between the two poles.
4. You should change this or that and see what happens. Leap!

This Page Intentionally Left blank

Session 4 - Time-of-Flight Principles



Basic Equations

Simple Linear Time-of-Flight Equations

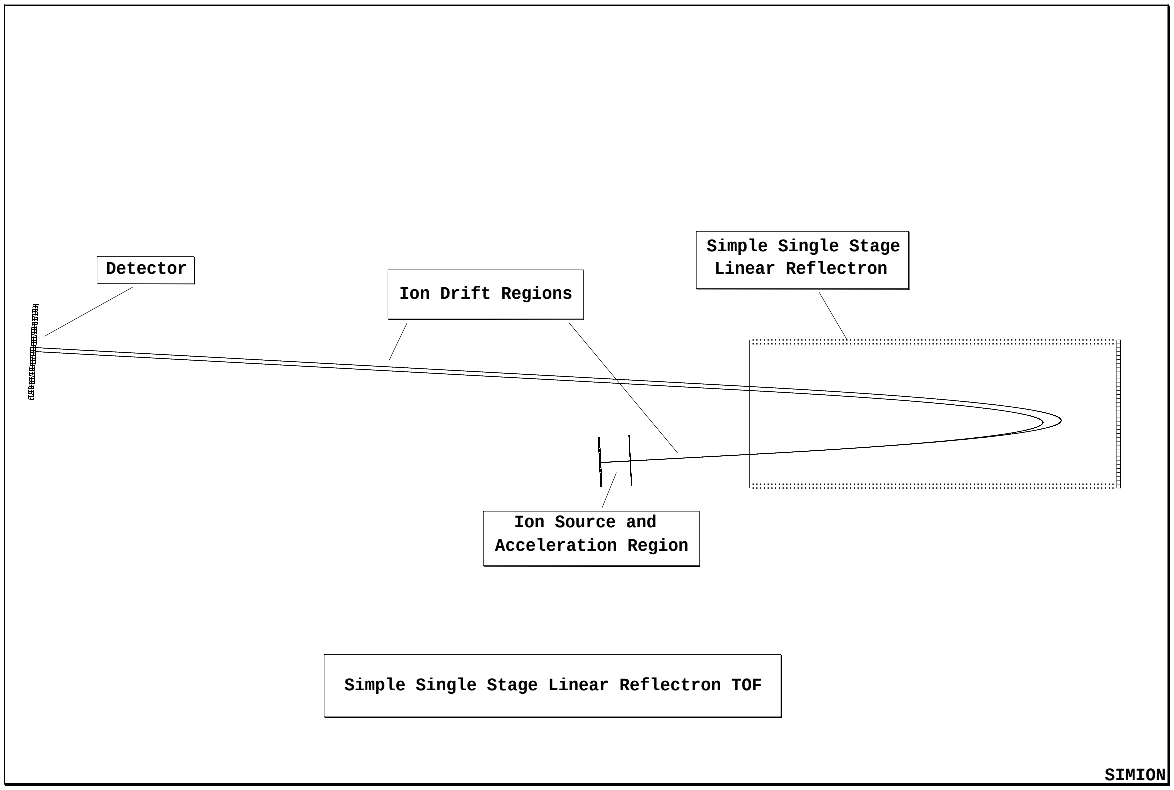
$$KE = mv^2/2$$

$$v_{mm/usec} = 13.891386\ (KE_{eV} / m_{amu})^{1/2}$$

$$t = d / v$$

$$t_{usec} = d_{mm}\ (\ m_{amu} / KE_{eV})^{1/2} / 13.891386$$

Simple Reflectron TOF Equations



$$t_{total} = t_{drift} + t_{reflectron}$$

$$t_{total} = d_{drift}/v + 2\ v/a$$

$$a = f\ /\ m$$

$$f = work/distance = Volts/m$$

$$t_{total} = d_{drift}\ (m)^{1/2}\ /(2KE)^{1/2} + 2\ (m)^{1/2}\ (2KE)^{1/2}/F$$

$$t_{total} = m^{1/2}(k1/KE^{1/2} + k2\ KE^{1/2})$$

Session 4 - Lab on Time-of-Flight

This Lab's Files are found in Directory: C:\CLASS\SESSION4\LINEAR

Flying Ions in a Simple Linear TOF

This model is used to demonstrate a very simple linear time-of-flight system. The demo makes use of voltage gradient time focusing for ions that are created at various locations in the ion formation region. The model is also used to demonstrate the use of the trajectory quality parameter to get proper discontinuity detection.

Creating the Required Files After Installation (First Time Only)

a. Remove all PAs From RAM
b. Use New and click the Use Geometry Files button. Use the LINEAR.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS\SESSION4\LINEAR).
c. Use the Save function to save the potential array as LINEAR.PA#.
d. Use the Refine function on LINEAR.PA# to create the required files for the demo.

Lab Steps Assuming the Required Files Have Been Created

1. Remove all PAs From RAM
2. Use View to load LINEAR.IOB from directory C:\CLASS\SESSION4\LINEAR
3. Load the ions in the LINEAR.FLY file.
4. Set time markers on at 1.0 usec with their color set to 15 (so the ion's markers use the same color as the ion's trajectories).
5. Load the data recording file LINEAR.REC and turn data recording on.
6. Fly the ions singly with trajectory quality set to 2 (default value). Note the KE error. This value is too large. The ions must be missing a grid edge.
7. Re-fly the ions with increasing trajectory qualities (>2) until the KE error is around 10^{-6} or less. Notice that the error suddenly improves between two successive values of data quality. This is because the CV detection is now just sensitive enough to see the grid edge. Notice the improvement in the TOF times (on the data recording screen).
8. Notice that although the ions are created in different locations that they arrive at virtually the same time. However, the compensation is not perfect. Can you explain why? Are we dealing with problems with SIMION or basic physics?
9. Load and fly the ions in the LINEAR3.FLY file (25, 50, and 100 amu ions). Notice that all three ion masses bunch equally well.
10. Modify the data recording parameters so that the ion's velocity is recorded when it splats. Are the velocity ratios correct between ions of different masses? Are the ion's velocities correct (use the equations above and check array potentials.

This Lab's Files are found in Directory: C:\CLASS\SESSION4\REFLECT

Using the Single Stage Linear Reflectron Demo

This model is used to demonstrate a simple single stage linear reflectron time-of-flight system. The demo makes use of a single stage linear reflectron to time focus ions of the same mass but different kinetic energies.

No After Installation Files Need to be Created

1. Remove all PAs From RAM
2. Use View to load TOF.IOB from directory C:\CLASS\SESSION4\REFLECT
3. Load the ions in the TOF.FLY file.
4. Load the data recording file TOF.REC and turn data recording on.
5. Fly the ions grouped as dots. Note the arrival times on the data recording screen.
6. Click the PAs tab, set the instance to one (the reflectron), and then click the Fast Adjust button. Record the current reflectron potential (should be 1000 V). Now increase the reflectron potential to 1500 volts and re-fly the ions. What happened? Now try 900 volts. Which ions get there first (low or higher energy)? Now try 1000.7 volts.
7. Select the detector instance (instance 3) on the PAs option screen. Click the Align button (left edge of screen) to align the workbench coordinates with the detector. Now fly the ions. Zoom in and watch them hit the detector. The dot speed slider can be used. Also try the single step function (just to the right of the trajectory quality panel - the Stp button).
8. Now go in and split the ion energies (was 50eV between two ions in each group) to 25 eV delta for 3 ions in each group (increase the ions to 3 and change delta energy to 25 eV) for each of the two groups. Re-fly the ions. What does your time focusing look like now? Explain what you see - physics or SIMION?
9. Reload the TOF.FLY ion definitions file, delete the second ion group, and now load the TIMES.REC data recording file. Fly the two ions and use the data recording information to determine the ratio of drift time to time spent in the reflectron. The ratio should be 1.0 but it isn't! What do you think is the cause? Hint: The acceleration stage creates a time shift.
10. To eliminate the impact of the ion acceleration stage, set its fast adjust voltage to zero (instance 2 - 800 volts by default), and then set the first ion's KE to 801 eV. Now re-fly the ions. Now they are no longer time focused!
11. We can pull the source further back from the reflectron to help the time focus. Make sure the Align button is raised. Select instance 2 (the source instance). Now depress the Align button to align the workbench coordinates with the working origin of the source array instance.
12. Click the Edit button on the PAs options screen. Move the source back by entering a negative value for X_{wb} (Hint: -91 is pretty close). The advantage of using the align function is that it allows us to move the source along the ion's flight path. Thus no beam alignment problems.
13. Now re-fly the ions and check splat time-of-flights. They should be pretty close if you used -91.
14. Compare the drift time verses reflectron time ratios. They are very close to 1.0. If you fly an intermediate energy ion (e.g. 3 ions with 25 eV delta KE), its time ratios will be ***very*** close to 1.0.

Session 4

This Lab's Files are found in Directory: C:\CLASS\SESSION4\FOCUS

How Refraction Impacts Time-of-Flight

This lab illustrates the errors in TOF introduced by the use of refractive elements like an einzel lens. A flat plane of ions entering an einzel lens does not leave as a flat plane of ions. However, the lab also demonstrates a trick that is useful to know.

No After Installation Files Need to be Created

1. Remove all PAs From RAM
2. Use View to load EINZEL.IOB from directory C:\CLASS\SESSION4\FOCUS
3. Load and fly the ions in the EINZEL.FLY. Set time markers on a 1.0 usec. with marker color set to 15 (uses the ion's color for marker color). Fly the ions singly as dots (fast - slider to right).
4. Note that while the ions enter the lens as a vertical line of dots they do not leave the einzel that way. Which ions are leading and which ions are trailing. What type of einzel is this, accel or decel - use PE view as an aid.
5. Now fast adjust the center electrode to -250 volts and re-fly the ions. This is an accel einzel. Note that the ion distortion is reversed. Can you explain why the ion trajectories are time distorted the way they are?

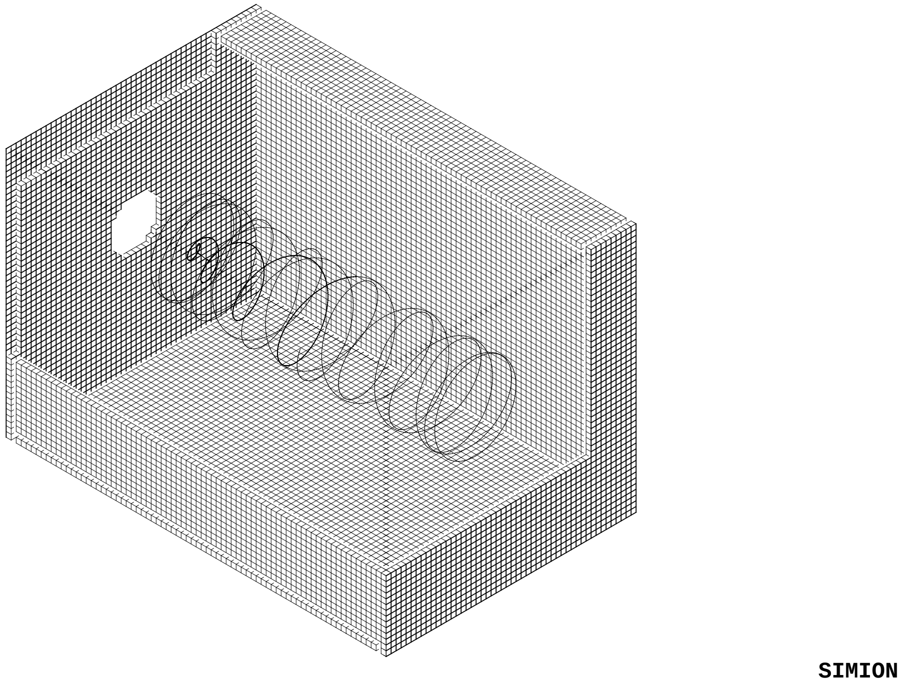
This leads to an idea! Why not use two einzels in series to focus the ions while compensating for the time distortion by making one einzel an accel and the other a decel. If you want to test your skill with instances, go to the main menu, load the EINZEL.PA0 file and save it as EINZEL.PB0 (a second separate einzel fast adjust file). Remove all PAs from RAM and reload the EINZEL.IOB. Stretch the workspace to the right (increase xmax). Now mark where you want the second einzel to be, click the PAs tab, now the Add button and select the EINZEL.PB0 potential array. Use instance editing to scale and position this instance properly. Now make instance 2 an accel einzel and fly ions. If you play around with voltages the time focusing problem can be pretty much eliminated.

If you're not that ambitious run the two einzel demo:

1. Remove all PAs From RAM
2. Use View to load FOCUS.IOB from directory C:\CLASS\SESSION4\FOCUS
3. Load and fly the ions in the EINZEL.FLY. Set time markers on a 1.0 usec. with marker color set to 15 (uses the ion's color for marker color). Fly the ions singly as dots (fast - slider to right).
4. Zoom into the group of dots near the focus point. Notice the that the time distortion is minimal.
5. Use various views to examine how the distortions are removed as the ions fly.

This page intentionally left blank

Session 5 -Ion Cyclotron Resonance Mass Spectrometry



Basic Equations

The Cyclotron Frequency Equation

cyclotron frequency (Hz) = f = qB/2□m

where:

q = charge*1.60217733 E-19

B = magnetic field in Tesla (T= 10^4 Gauss)

m = mass = amu*1.6605402 E-27

Note that the cyclotron frequency of an ion is inversely proportional to it's mass.

Session 5 - Ion Cyclotron Frequency Mass Spectrometer Lab

This Lab's Files are found in Directory: C:\CLASS\SESSION5\ICR

This lab is designed to demonstrate the trajectories of ions through an ion cyclotron resonance mass spectrometer. The model utilizes a three dimensional rectilinear ICR cell and a uniform magnetic field. User Programs allow adjustment of the ion parameters and operating mode of the ICR. The workbench contains two overlapping instances: The electrostatic ICR cell (instance), and the magnetic field (instance). User programs randomize the ion initial conditions and allow the tuning parameters to be adjusted by the user during ion flying. The randomly assigned starting energies, positions, and directions are bounded by conditions set by the user.

Creating the Required Files After Installation (First Time Only)

a. Remove all PAs From RAM

b. Use New and click the Use Geometry Files button. Use the ICR.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS\SESSION5\ICR).

c. Use the Save function to save the potential array as ICR.PA#.

d. Use the Refine function on ICR.PA# to create the required files for the demo.

Lab Steps Assuming the Required Files Have Been Created

Single Ion Trajectory

1. Remove all PAs from RAM. Use View to load the file ICR.IOB from the directory c:\class\session5\ICR; reply Yes to "Restore potentials to the value when IOB was saved?".
2. Load the trajectory file ICR_1.FLY. This will fly a single ion of mass 1000. The ion will have a randomly selected starting position and angle, set via user adjustable variables in the User program.
3. Select trajectory quality = 0, ion as lines. Be sure rerun is not activated.
4. Fly the ion; accept all the default values that appear on the user adjustable variables screen that appears except for the Delay_Start, which should be set to 20 (push the Fly'm button on this window). The ion will start out and then an Rf field will be applied to the ICR cell to pump up the kinetic energy of the ion and get it moving at it's cyclotron frequency. The ion will change from blue to red when the Rf is initiated, and from red to green when it is turned off. Stop the trajectory calculation after a brief period and view the trajectory from different orientations. You can restart the ion several times at different orientations and get a good feel for what is happening to it. The ion moves back and forth between the trapping plates (end plates) while spiraling around the axis of the magnetic field.

Pumping Kinetic Energy into the ions - the Rf phase.

5. Select the XY orientation and perform a 3D-Zoom of the area (-14,9) to (80,-9). Now switch to the XZ orientation and select the PE View. Start the ion flying and watch the potential energy surface as it varies with the Rf voltage. This is how the ion acquires the kinetic energy. When the pumping period ends, the potential is defined by the trap plate potentials, which keep the ion from wandering out of the cell.

Getting the Ions in Phase - groups of ions and separation by mass.

In order to detect the ions they must be moving in phase inside the ICR cell. The Rf pumping gives them the kinetic energy, and their mass determines their cyclotron frequency. This example demonstrates that ions of different mass are separated based on their cyclotron frequency.

6. Load the trajectory file ICR_3.FLY; set the parameters to grouped, dots, slider to the right, rerun on, quality 0. Start a Fly'm; In the Adjustable Variables window push the As Defined button on the top row to reset all of the variables. Fly the ions.
7. Select the XY orientation and watch as the ions are started, when they turn to red switch to the ZY orientation and watch as the two different groups are separated out. Note that one group is rotating at a higher frequency than the other. Which group should have the higher cyclotron frequency?

Measuring the Cyclotron Frequency with SIMION

In this example the theoretical cyclotron frequency will be compared against the SIMION calculated frequency. A single ion's cyclotron frequency will be measured using markers and data recording.

7. Load the trajectory file ICR_W.FLY. In the trajectory definition screen at the bottom select Define in the Data Recording section.

   a. At the top of the data recording window push the button marked Blank.

   b. Push the button labeled TOF in the top section, and the button marked Y= in the middle section. The time of flight (TOF) will be recorded each time the ion passes through the Y=0 plane.

   c. On the bottom left select Verbose, and then at the top left select OK.

   Push down the Record button on the bottom line of the ion definition screen, and select OK.

8. Set the trajectory parameters to lines, rerun off, quality 0, ZY orientation. On the lower left push down the Vew RD button so it is red. Start the ion, in the adjustable variable screen push the button labeled E Fmt so it is black and then set the parameters as shown below (be sure to note changes in the start and end mass):

Bx_Gauss	30,000
Percent_Energy_Variation	0
Cone_Angle_Off_Axis	0
Random_Offset	0
Random_TOB	0
Start_Rht_Voltage	0
Capture_Voltage	0
Rf_Voltage	20
Starting_Mass	1950
Ending_Mass	2050
Rf_Sweeptime	120
Start_Delay	1
Update_PE	1

9. Fly the ion. After it has made 5-6 full revolutions in the green phase stop the ion. Zoom in on the markers, are they all perfectly aligned on the Y=0 axis? Why not? The variation in the time step at quality level 0 results in the marker (and data recording) occurring not exactly at Y=0, but

within the time step that includes Y=0. Thus the variation is dependent upon the length of the time steps as the ions approach the Y=0 boundary.

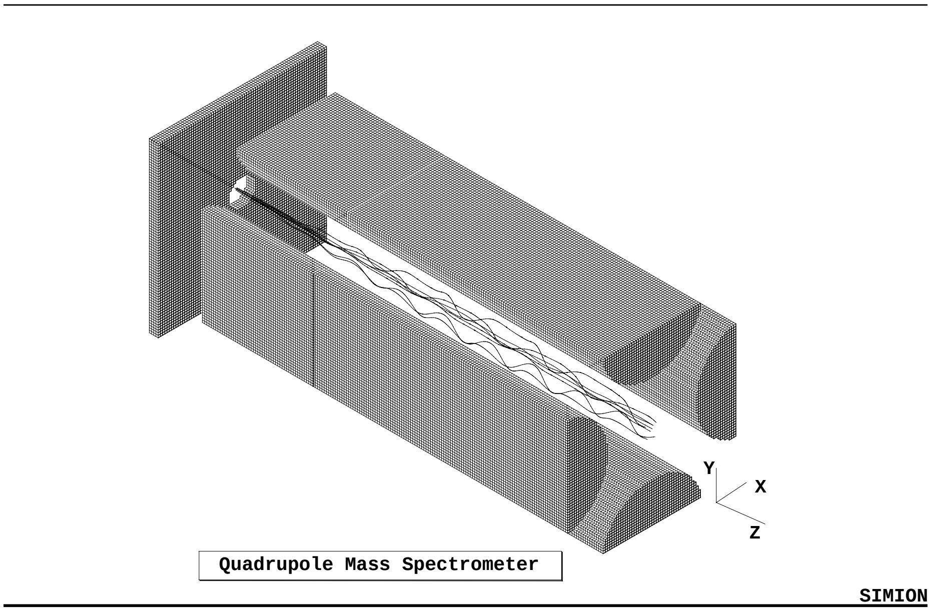
10. Calculate the period for one revolution (using the numbers on the recording window). The theoretical cyclotron frequency for a 2000 amu ion in a 30,000 gauss field is 23.03417 kHz, how does this compare with the frequency measured using time markers? Calculate the ratio of Theoretical/SIMION.
11. Set the quality to 2 and rerun the trajectory. Zoom in on the time markers, do they all align now? Why? With the quality level at 2 the program performs a binary boundary approach to the Y=0 boundary. Recalculate the period and cyclotron frequency using the times in the data recording window. How does this compare to the Theoretical value.
12. Rerun the ion, in the Adjustable Variable screen change the Bx_Gauss to 60,000 (double its original value). Now what is the cyclotron frequency, how did this affect the radius? How would you use data recording to get a measurement of the radius?

Trapping ions formed outside the ICR cell

This example illustrates a method for trapping ions created outside the cell. Problems in trapping ions of widely varying mass due to their velocity differences are illustrated.

13. Load the ICR_4.FLY trajectories. Set to dots, grouped, slider to the right, quality 0, rerun on, XY orientation, and fly the ions. Use the As Defined user adjustable variable values and fly the ions. They will change color when the pump field comes on and when it is turned off. This works nicely for trapping these ions. What will happen when we have ions of widely varying mass?
14. Load the ICR_5.FLY trajectories and fly them as grouped dots with quality of 0, rerun on. The ions have been started with the same approximate kinetic energy (as set by the user defined variable). Note that the heavier ions (red) are not moving as fast as the lighter ions (blue), and that some of the blue ions enter and leave the cell before the trapping voltage is applied. How would you solve this problem?

Session 6 -Quadrupole Principles



Basic Equations

The Quadrupole Field Equations

$$d^2x/dt^2 + 2e/mr_o^2(U - V\cos w_f t)x = 0$$
$$d^2y/dt^2 + 2e/mr_o^2(U - V\cos w_f t)y = 0$$
$$d^2z/dt^2 = 0$$

$2r_0$ = distance between opposing rods

w_f = angular frequency of applied field

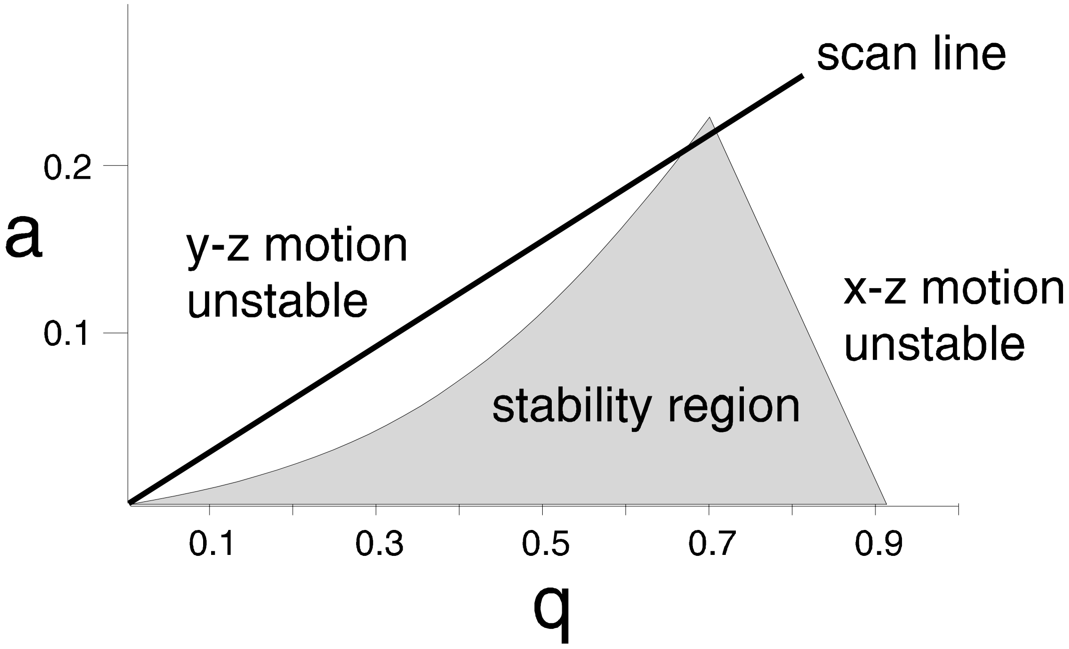
$$r_{\text{circular rods}} = 1.1487r_o$$

Determination of U and V via a_n and q_n

$$a_x = -a_y = 8eU/mw_f^2r_o^2$$
$$q_x = -q_y = 4eV/mw_f^2r_o^2$$

Session 6 - Quadrupole Lab

Quadrupole Stability Diagram



This Lab's Files are found in Directory: C:\CLASS\SESSION6\QUAD

This lab is designed to demonstrate the trajectories of ions through a quadrupole mass spectrometer, and to illustrate the effects of tuning parameters on the trajectories and the transmission of the ions through the quadrupole. The model utilizes User Programs and the workbench feature of SIMION. It contains three different overlapping instances (pieces of the quadrupole) that are synchronized in their operation: The quadrupole lens (a 3D array), the quadrupole (a mirrored 2D array), and the exit lens with a plate detector (a 3D array). User programs randomize the ion initial conditions and allow the tuning parameters to be adjusted by the user during ion flying. The randomly assigned starting energies, positions, and directions are bounded by conditions set by the user.

Creating the Required Files After Installation (First Time Only)

a. Remove all PAs From RAM
b. Use New and click the Use Geometry Files button. Use the QUAD.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS\SESSION6\QUAD).
c. Use the Save function to save the potential array as QUAD.PA#.
d. Use the Refine function on QUAD.PA# to create the required files for the demo.

e. Remove all PAs From RAM
f. Use New and click the Use Geometry Files button. Use the QUADIN.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS\SESSION6\QUAD).
g. Use the Save function to save the potential array as QUADIN.PA#.
h. Use the Refine function on QUADIN.PA# to create the required files for the demo.
i. Remove all PAs From RAM

j. Use New and click the Use Geometry Files button. Use the QUADOUT.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS\SESSION6\QUAD).

k. Use the Save function to save the potential array as QUADOUT.PA#.

l. Use the Refine function on QUADOUT.PA# to create the required files for the demo.

Lab Steps Assuming the Required Files Have Been Created

1. Remove all PAs from RAM. Use View to load the file QUAD.IOB from the directory c:\class\session6\group; reply Yes to "Restore potentials to the value when IOB was saved?".
2. Load the trajectory file QUAD.FLY. This will fly 25 ions of mass 100, the mass the quad is tuned to pass through. The ions will have randomly selected starting positions and angles, set via user adjustable variables in the User program.
3. Select trajectory quality = 0, ions as lines, ungrouped. Be sure rerun is not activated.
4. Fly the ions; accept all the default values that appear on the user adjustable variables screen that appears (push the Fly'm button on this window).
5. View the ion trajectories using each view orientation, XY, ZY and XZ. **In the XY view note the nodes that appear in the trajectories.** In the XY view make a 3D zoom from X=5 to X=85 (leaving out the inlet and outlet optics), switch to ZY view and look at the trajectories.
6. In the ZY view make a cut that includes only the bottom and two side rods, switch to the 3D ISO view. Note the regions on the electrodes where the first instance interfaces with the second (the long quad rods), and where the second instance (the long quad rods) interfaces with the exit region.
7. Exit from View (use the Quit button on the left). In the main menu screen select the QUADIN.PA0 array (push the button so it turns red), and then select Modify. This takes you into the modify screen where you can see how the geometry was defined. Select the XYZ view, note that is a planar array mirrored in Y and X. This saves array space (and memory!).
8. Leave Modify **using the Quit button.** Select the array QUAD.PA0 and look at it in Modify. What symmetries are used? Why? This is a simple 2D array, lots of space is saved by taking advantage of the symmetries of this system.
9. Leave Modify **using the Quit button**. Select the array QUADOUT.PA0 and look at it in Modify. What symmetries are used here?
10. Leave Modify **using the Quit button**.
11. Push the View button to return to the View screen. Push the PA tab and cycle through the instances (upper left, blue panel), note they correspond to the three files you looked at in Modify. Instances can be added or deleted from the Workbench using the buttons on this window.

Mass Filtering - where do the ions go?

12. Load the trajectory file QUAD1.FLY. This file contains three sets of ions, one of mass 95 (below the quad tune point, color red), one set at mass 100 (the quad tune point, blue), and a third set at mass 105 (green). Again the ions are started at random angles and starting positions within the bounds set by the User Adjustable variables. Fly the ions and observe the trajectories.
13. View the trajectories in the ZY view, note which set of quadrupole rods the lighter ions impact, and which rods the heavier ions impact. Look at this from different views. Do any of the lighter or heavier ions make it to the detector?
14. One of the User Adjustable variables in the quadrupole axis voltage; this can be set on the screen that appears after the Fly'm button is first pushed. Note the value (-8 volts) and run a set of trajectories. **Write down the approximate positions in X where the light and heavy ions hit the rods.** Now run the trajectories over and set the quadrupole axis voltage to -24 volts. How did this affect where the ions hit the rods, did any of the ions make it to the detector? What does this tell you about how the axis voltage may affect the peak resolution, the peak shape, and the signal intensity.
15. Load in the QUAD.FLY trajectory file and fly the ions with the quadrupole axis voltage set to -8 volts (default value), and in the XY view note the nodes that appear in the trajectories. Now re-fly the ions with this voltage set to -2 volts. How did this change the general shape of the trajectories,

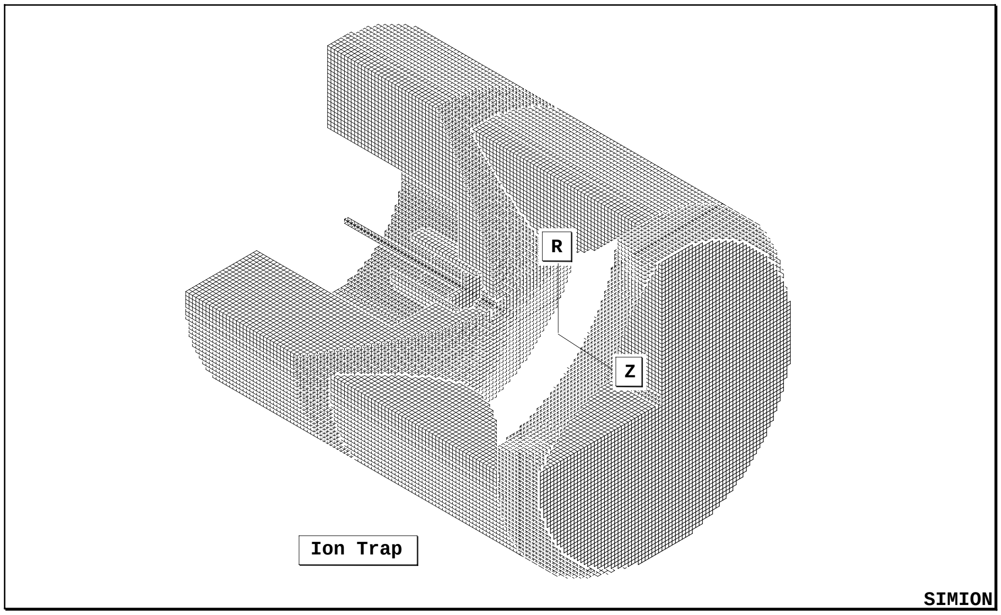
how many nodes appear now? Try other values of this voltage, what happens if this voltage is set positive? Could this be used to estimate the initial kinetic energy of the ions?

Ion Optics - how to get the ions into the quadrupole.

The effect of the ions' trajectories as they enter the quad on the ion transmission is demonstrated. Dynamic display of potential fields is shown, and the manner in which SIMION handles trajectories at interfaces between different instances is demonstrated.

16. Load the QUAD3.FLY trajectory file and fly the ions with the default User Adjustable values (hint, select the As Defined button). Now re-fly the ions with the User Adjustable variable Cone_Angle_Off_Vel_Axis 15 degrees. What affect does this have, notice that now many ions are lost. The acceptance ellipse for the quad is finite and small, ions that enter outside of this are lost.
17. Load the QUAD1.FLY file. Run the ions as dots with the speed slider all the way to the right, grouped, with rerun on, be sure trajectory quality is set to 0. View in the XY and YZ planes.
18. Switch to the XY view and make a cut from X=0, Y=0 to X=10, Y=-9. Switch to the XZ orientation and select PE View (potential energy surface). Set the drawing quality to 0 and the Gstep to 5 or higher. Fly the ions and watch the PE surface undulate. Switch back to the WB view, select ZY orientation, switch back to PE view, and watch again as ions fly.
19. Stop the Fly'm. Switch back to WB view, select XY orientation, zoom out one level and now make a second cut so you have the region (0,0) to (18,-9) zoomed into. Switch to XZ orientation and select the PE view and fly the ions. Notice that only the left side of the field is changing, the right side is staying constant. What happens when the ions finally get into the field on the right hand side; what happens when there are no longer any ions in the left hand region? Recall that the workbench has three instances in it, making up the quad. When does the program perform calculations for each instance? (Only when there are ions within the instance.) What happens when ions are in two instances?
20. Experiment with the effects of various tuning parameters (inlet and exit voltages etc), ion parameters (increase the energy spread, the position where ions are born) via the User Defined Variables.

Session 7 - Ion Trap Principles



Basic Equations

The Ion Trap Field Equations

$$d^2z/dt^2 - 4z(U - V\cos w_f t)e/mr_o^2 = 0$$

$$d^2r/dt^2 - 2r(U - V\cos w_f t)e/mr_o^2 = 0$$

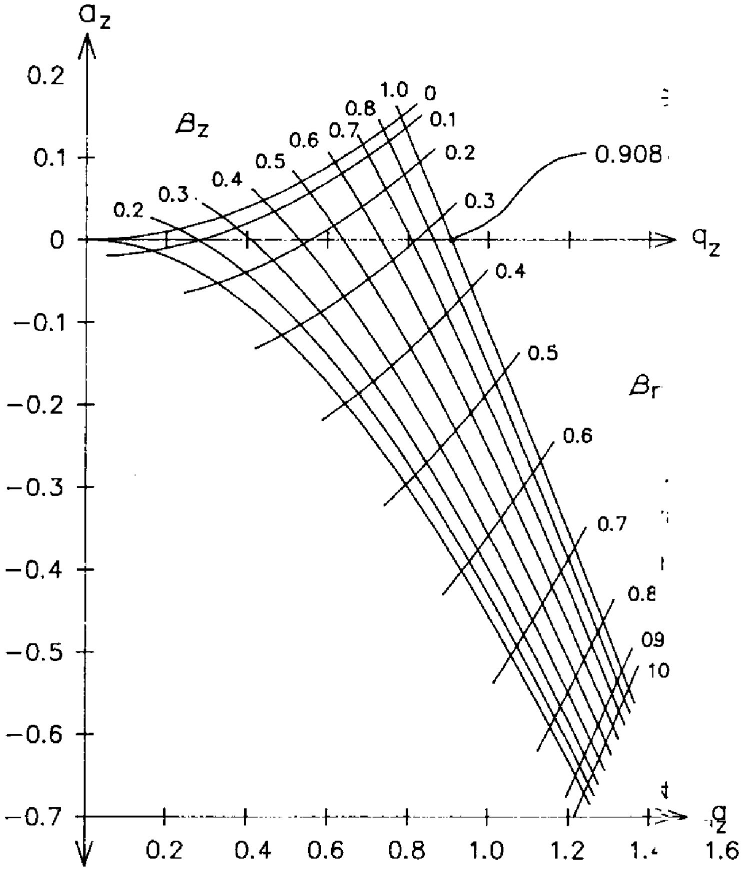
r_0 = radius of trap
w_f = angular frequency of trap

Determination of U and V via a_n and q_n

$$a_z = -2a_r = -16eU/mw_f^2r_o^2$$

$$q_z = -2q_r = -8eV/mw_f^2r_o^2$$

Ion Trap Stability Diagram



Secular Ion Motion Frequencies

$$w_r = B_r w_f/2$$
$$w_z = B_z w_f/2$$
$$w_f = \text{trap frequency}$$

Values for B_u in terms of a_u and q_u

$$B_u = (a_u + z_u^2/2)^{1/2} \text{ for } q_u < 0.4$$

or in complete form

$$B_u^2 = a_u + q_u^2/((2+B_u)^2 - a_u - q_u^2/((4+B_u)^2 - a_u - q_u^2/(\ldots\text{etc.})))$$
$$+ q_u^2/((B_u - 2)^2 - a_u - q_u^2/((B_u - 4)^2 - a_u - q_u^2/(\ldots\text{etc.})))$$

Session 7 - Lab on Ion Trap

This Lab's Files are found in Directory: C:\CLASS\SESSION7\GROUP

Flying Ions in an Ion Trap

The first part demonstrates the trajectories of a single ion within a trap. The user programs allow the user to change the operating point of the trap (base mass, a_z, and q_z) as the ion flies.

Creating the Required Files After Installation (First Time Only)

a. Remove all PAs From RAM
b. Use New and click the Use Geometry Files button. Use the TRAP2.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS\SESSION7\GROUP).
c. Use the Save function to save the potential array as TRAP.PA#.
d. Use the Refine function on TRAP.PA# to create the required files for the demo.

Lab Steps Assuming the Required Files Have Been Created

1. Remove all PAs From RAM
2. Use View to load TRAP.IOB from directory C:\CLASS\SESSION7\GROUP
3. Load and fly the single ion in the SINGLE.FLY ion group definition file (trajectory quality set to 0 for speed). Fly the ion as both a line and as a dot (use slider to vary dot speed). **Notice that the ion has a different trajectory on every flight**. This is because the user program randomizes the ion's starting parameters automatically!
4.. While an ion is flying, click the AdjV tab to gain access to the user program's adjustable variables. Change the _AMU_Mass_per_charge variable from its initial value of 40 to 90 then 100 and then 101 (the ion has a mass of 100). What happens ?
5. Use the _Az_tune and _Qz_tune parameters to move the ion about in the stability diagram. Can you make the ion hit the ring electrode? How?
6. Use data recording to obtain the KE of the ion when it splats. Now fly 10-20 ions off the right edge of the stability diagram. Increase the mass of the ions to 500 and repeat. How does the ion's mass impact its ejection energy?

The second part demonstrates interactions between ions flying in a trap. Factor charge repulsion as well as viscous cooling are used to demonstrate the formation of ion crystal patterns within the trap.

7. Load and fly the ten 100 amu ions in the GROUP.FLY file. Fly them as dots (fast - slider to right), grouped, with trajectory quality set to zero. Hit the Esc key to stop the flight. Re-fly the ions. Notice that they have different trajectories in every flight.
8. Now re-fly the ions with factor repulsion set to 1.0 (each ion just represents itself). The ion motions now include the effect of charge repulsion of the ions.
9. Now, while the ions are flying with charge repulsion active, click the AdjV tab to gain access to the user program's adjustable variables while the ions are flying. Change the _Linear_Damping to 1.0. Notice that the trajectories collapse (ions are now being cooled). Zoom in. View the ion motions from the side and end-on. Notice that the 10 ions have formed a crystal pattern. Use dots and lines to see ion motions and streak patterns. Using a 3D volume zoom can be very helpful here. Change the factor repulsion to 1,000 so that each ion represents a cloud of 1,000 ions and see the change in the pattern size.
10. Load and fly the ions defined in GROUP.FLY file with factor repulsion set to 1.0 and linear damping set to 1.0. There are 45 ions defined (15 -100 amu **red**, 15 - 200 amu **green**, and 15 - 300 amu **blue**). Notice the crystal pattern they form. Which ions are on the inside which ions are on the outside? Now change the _AMU_Mass_per_charge variable from 40 to 100 while the ions are flying. Notice the dramatic change in the edge view of the crystal pattern. Now bump the value to 102 and watch the 100 amu ions exit and the crystal reform. Interesting.

Session 7

<u>This Lab's Files are found in Directory: C:\CLASS\SESSION7\TICKLE</u>

<u>Using the End Cap AC Voltages to Excite Selected Ions</u>

If the end caps have a tickle voltage that has the secular frequency w_z of specific ions' mass for that a and q value the ions will gain energy and tend to leave the trap through the end cap holes. The user programs in this demo compute the proper tickle frequency and allow the user to change the tickle voltage and other parameters while the ions are flying. This demo also makes use of a collisional cooling model that uses mean free path and the mass of the cooling gas to simulate random collisional cooling of the ions.

<u>Creating the Required Files After Installation (First Time Only)</u>

a. Remove all PAs From RAM
b. Use New and click the Use Geometry Files button. Use the TRAP2.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS|SESSION7\TICKLE).
c. Use the Save function to save the potential array as TRAP.PA#.
d. Use the Refine function on TRAP.PA# to create the required files for the demo.

<u>Lab Steps Assuming the Required Files Have Been Created</u>

1. Remove all PAs From RAM
2. Use View to load TRAP.IOB from directory C:\CLASS\SESSION7\TICKLE
3. Load and fly the ions in the TICKLE1.FLY (trajectory quality set to 0 for speed). Fly the ion as grouped as dots (fast - slider to right). One group of ions (black) are not affected (mass not in resonance). However, the other group of ions (blue - red) will eventually be blown out of the trap. Notice that these ions change from blue to red. The user program does this to indicate the polarity of the tickle voltage being placed on the end caps. This shows that forcing being applied to the ions.
4. Load and fly the ions in the TICKLE.FLY file (resonance ions only). Increase the tickle voltage and re-fly the ions.
5. Change the _AMU_Mass_per_charge variable from 40 to 80, re-fly and see what happens. Try flying ions with this variable set to 10 amu. Can you explain what you see?
6. Change the _AMU_Mass_per_charge variable from to 303, tickle to zero volts, and re-fly. Now turn on 2.0 volts of tickle while the ions are flying. This is an example of using tickle on the right stability edge to help eject ions.
7. This model makes use of a collisional cooling model based on mean free path assumptions controlling random collisions with a bath gas of a given mass (helium - 4 by default). Change the mean free path to 1 mm and see what effect increased gas pressure has on the tickle. **Note setting mean free path to zero will turn off collisional damping**.
8. You can also use the TRAP1.GEM file to create a new TRAP.PA# file with hole in left end cap and a grid over the hole in the right end cap. Save the array, refine it, remove all PAs, reload the TRAP.IOB with View, and examine the impact of hole symmetry in the end caps on the preferred end cap for ion exit of the trap.

This Lab's Files are found in Directory: C:\CLASS\SESSION7\INJECT

Injecting Ions into an Ion Trap

This demo models the injection of ions into an ion trap. 100 Ions are injected singly and the user programs perform a Monte Carlo analysis of each ion's fate. The model makes use of collisional cooling with mean free path and the mass of the cooling gas. It also includes an estimate of the reduction of the mean free path as a function of the ion's mass (and assumed complexity - cow pie effect):

mfp_{est} = mfp/(1 + (n-1) _Avg_Visibility)

Where:

n = ion mass / _Avg_Mass_per_atom
_Avg_Mass_per_atom = 12 (default)
_Avg_Visibility = 0.666 (default)

Creating the Required Files After Installation (First Time Only)

a. Remove all PAs From RAM
b. Use New and click the Use Geometry Files button. Use the TRAP1.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS|SESSION7\INJECT).
c. Use the Save function to save the potential array as TRAP.PA#.
d. Use the Refine function on TRAP.PA# to create the required files for the demo.

Lab Steps Assuming the Required Files Have Been Created

1. Remove all PAs From RAM
2. Use View to load TRAP.IOB from directory C:\CLASS\SESSION7\INJECT
3. Load and fly the ions in the INJECT.FLY (trajectory quality set to 0 for speed). Fly the ion as singly as dots (fast - slider to right). Fly about 30 ions and hit Esc to quit. Note the number of ions actually trapped (from the data screen).
4. The mean free path is set to 40 mm (approx 10^{-4} torr). Decrease the mean free path to 4 mm and re-fly the ions. Are the results what you expected?
5. Now change the mass of the ions from 100 (their default) to 500, mean free path to 40, and rerun 30 or so ions. How many did you trap? Not too many! Now re-fly these ions with the _AMU_Mass_per_charge variable set to 60 amu. Compare the results. Can you explain what you see?
6. You can also change the sample and focus ring potentials and see their impact on ion capture. Give the ions a lot of energy (sample potential) and see what happens.
7. You can also use the TRAP2.GEM file to create a new TRAP.PA# file with ideal grids across the end cap holes. Save the array, refine it, remove all PAs, reload the TRAP.IOB with View, and examine the impact of grids. The options are endless.

<u>This Lab's Files are found in Directory: C:\CLASS\SESSION7\MSMS</u>

<u>Simulation of MSMS Fragmentation an Ion Trap</u>

This demo models the use of tickle on the end caps to fragment a parent ion into a daughter ion. 100 Ions are injected singly and the user programs perform a Monte Carlo analysis of each ion's fate. The model makes use of collisional effects of helium to fragment the parent ion. The model assumes this is a heating process that accumulates energy (most individual collisions are less than 0.1 eV). A fraction (user definable - assumed 0.5) of the collisional KE is assumed to be retained as heating of the parent ion. When this accumulated heating energy exceeds a designated value (10. eV by default) the ion is fragmented into its daughter ion. If the daughter ion is retained for 20 rf cycles (default) it is assumed to have been trapped.

<u>Creating the Required Files After Installation (First Time Only)</u>

a. Remove all PAs From RAM
b. Use New and click the Use Geometry Files button. Use the TRAP2.GEM file by pointing to its button and clicking both mouse buttons (in C:\CLASS|SESSION7\MSMS).
c. Use the Save function to save the potential array as TRAP.PA#.
d. Use the Refine function on TRAP.PA# to create the required files for the demo.

<u>Lab Steps Assuming the Required Files Have Been Created</u>

1. Remove all PAs From RAM
2. Use View to load TRAP.IOB from directory C:\CLASS\SESSION7\MSMS
3. Load and fly the ions in the MSMS.FLY (trajectory quality set to 0 for speed). Fly the ion singly as dots (fast - slider to right). Fly about 30 ions and hit Esc to quit. Note the number of daughter ions actually trapped (from the data screen).
4. The mean free path is set to 40 mm (approx 10^{-4} torr). Decrease the mean free path to 4 mm and re-fly the ions. Are the results what you expected?
5. Change the _AMU_Mass_per_charge variable from 40 to 10, mean free path = 40 mm, re-fly and see what happens. Try flying ions with this variable set to 60 amu. Can you explain what you see?
6. You can also use the TRAP.GEM file to create a new TRAP.PA# file with open end cap holes (both end caps). Save the array, refine it, remove all PAs, reload the TRAP.IOB with View, and examine the impact of holes in the end caps on ion stability.

Session 9 - Geometry Files

What is a Geometry File

- A geometry file is an ASCII file with the .GEM filename extension that contains the electrode/pole definitions for a potential array.
- It also may contain the definition of the potential array:

pa_define(100,20)

SIMION Functions that Use Geometry Files

- The New function via the Use Geometry File button (*makes use of pa_define commands*).
- The Modify function via the GeomF button (*ignores pa_define commands for an existing array*).

How Geometry Files Work

- The geometry files contain a collection of fill commands.
- Fill commands can define complex areas/volumes through the use of inclusion/exclusion commands (e.g. within and notin), orientation commands (e.g. locate), and large collection of basic shape commands (e.g. circle).
- Each successive fill command has increasing priority. Later fill commands override earlier fill commands.
- SIMION makes use of a geometry compiler to create an ordered fill search tree.
- Each point within the array is checked to see if it is affected by a fill command (from last fill toward first fill).

The Geometry File Language

- The Language is nested - commands fit inside each other.

```
e(1){fill{within{circle(0,0,5)}notin{box(0,0,2,2)}}}
e(1)                                ; electrode of one volt
    {
    fill                            ; fill command
        {
        within{   circle(0,0,5)}    ; within a circle at 0,0  r = 5
        notin{    box(0,0,2,2)}     ; notin a box with corners of
        }                           ; x,y:   0,0 and 2,2
    }
```

- Each command has its scope: The region of commands that are under its influence.
- Commands can have parameters and/or scope
- Inline comments begin with a semicolon.

The Logic Used in Fills

- A point will be within a fills volume if it is contained in at least one of the fill’s within volume and not contained in any of the fill’s notin volumes.

```
e(1)                                ; electrode of one volt
    {
    fill                            ; fill command
        {
        within{   circle(0,0,5)}    ; within a circle at 0,0  r = 5
        within{   circle(20,0,5)}   ; within a circle at 20,0  r = 5
        notin{    box(0,0,2,2)}     ; notin a box with corners of
                                    ; x,y:   0,0 and 2,2
        notin{    box(20,0,22,2)}; notin a box with corners of
        }                           ; x,y:   20,0 and 22,2
    }
```

- A point satisfies a within or notin if it is contained in the intersection volume defined by the shape commands within its scope.

```
within{circle(0,0,5) circle(5,0,5)}      ; within intersection area

notin{box(0,0,5,5) circle(0,0,5)}        ; notin first quadrant
```

2D and 3D Considerations

- All 2D shape commands (e.g. box and circle) are defined in x and y and have a +- 10^6 extent in the z direction when applied to 3D arrays.
- All 3D shape commands (e.g. sphere) have their 2D intersection area (z = 0) when applied to 2D arrays.

Location, Scaling, and Orientation

- Locate commands are used to locate, scale, and orient anything at any level. **<u>They even can be nested within themselves:</u>**

```
locate(100)
    {
    fill
        {
        within{sphere(0,0,0,50,60,40)}
        notin{sphere(0,0,0,45,55,45)}
        locate(0,0,0,1,0,0,-90)
            {
            notin{circle(0,0,10,5)}
            }
        }
    }
```

- Example above creates ellipsoidal shell with an elliptical hole vertically up through the center.

The PA_Define Command

- This command defines the potential array to use.
- If it appears in a .GEM file it must be the first command.
- It is used by the New command and ignored by the Modify Command

```
PA_Define( nx, ny, nz,        ; if nz = 1 then 2D array
           Symmetry,          ; Planar or Cylindrical
           Mirroring,         ; legal combinations of x,y,z, or none
           Type,              ; Electrostatic or Magnetic
           ng)                ; Value of ng parameter
```

The Geometry File Development System

- Accessed from within Modify with the GeomF button.
- Geometry files can be edited, test compiled and debugged.
- Potential array can be erased (electrode/pole points removed).
- Geometry definitions can be inserted into potential array.
- Allows full manual control over translation, scaling and orientation via locate parameter panels (serves as outermost Locate command).

Session 9 - Lab on Geometry Files

This Lab's Files are found in Directory: C:\CLASS\SESSION9\TRAP

Inserting a Geometry File into Different Sized Arrays

The contents of a geometry file can be scaled to fit within arrays of different sizes. This lab shows how this is done

No After Installation Files Need to be Created

1. Remove all PAs From RAM
2. Use New to load TRAP.GEM from directory C:\CLASS\SESSION9\TRAP
3. Examine the array with View and then Modify.
4. The Array is 399x by 173y. Use Modify to halve the size of the array (200x by 87y);
5. Click the GeomF button to access the geometry file development system.
6. Test compile the geometry file by clicking the Compile button.
7. Change the Scale panel from 1.0 (default) to 0.5 to re-scale the geometry definitions to the smaller potential array.
8. Click the Erase Entire PA button then click the Insert into PA button. You should now be looking at the halved scale potential array.
9. Follow the above steps 4-8 with the appropriate numbers to halve the array again. What can you say about the change in resolution. Which array would you expect to model an ion trap more accurately?
10. Now we are going to insert this geometry file into a bigger array. In Modify, try to set a new array size of 797x by 345y. Did you have problems? What do you think is wrong? How would you fix it?

11. Remove all PAs From RAM
12. Use New to define a cylindrical electrostatic 2D array with 797x by 345y coordinates (you will have to increase the size of the Max PA to get these values).
13. Now enter Modify and Click the GeomF button.
14. Change the Scaling to 2.0 to double the geometry size to fit the new array.
15. Click the Erase Entire PA button then click the Insert into PA button. You should now be looking at the double scale potential array.
16. Click the Keep button and enter View. There are a lot of points in this array. Change drawing quality to 9 (click Esc if drawing takes too long).

<u>This Lab's Files are found in Directory: C:\CLASS\SESSION9\TRAP</u>

<u>Modifying the Trap Geometry File into a Stretched Trap</u>

The contents of the trap geometry file above will be modified to define a Finnigan stretched trap which will be saved as the STRETCH.GEM geometry file

<u>No After Installation Files Need to be Created</u>

1. Remove all PAs From RAM
2. Use New to load TRAP.GEM from directory C:\CLASS\SESSION9\TRAP
3. Examine the array with Modify.
5. Click the GeomF button to access the geometry file development system.

6. Click the Edit button to edit the geometry file with EDY.
7. Rename the File STRETCH.GEM
   EDY keystrokes: **Esc N STRETCH.GEM** and press **Enter**
8. Now Let's save the file, exit EDY and Select and use this file as the geometry file:
   Esc Q Y and your back in SIMION.
9. Click the Select button and click both mouse buttons on STRETCH.GEM to make it the currently selected .GEM file.
10. Use the Erase all points button and then the Insert into PA button to verify that the geometry file still makes the array OK.

11. Now the task is to shift both end caps 0.76 mm away from the trap center to create a stretched trap. The nominal scaling of this potential array is 0.1 mm/grid unit. Thus we need to shift the left end cap (and its contents) 7.6 grid units to the left and the right end cap 7.6 grid units to its right.

    The best strategy is to insert locate commands that have the proper shifts and enclose the proper electrode definitions. This will result in the caps being shifted properly.

    <u>Go for it!</u>

12. Assuming you've given it a good college try and failed. Use the Select button and click both mouse buttons on the STRETCH1.GEM file. Insert the geometry from this file. It works! Look at the file with EDY to see how it was done.

This Lab's Files are found in Directory: C:\CLASS\SESSION9\EINZEL

Creating a geometry file for an EINZEL lens

A geometry file for an einzel lens will be created.

No After Installation Files Need to be Created

1. Remove all PAs From RAM
2. Load the EINZEL.PA# array with Load
3. Look at the electrode definitions with Modify.
4. Write down, on paper, the image of a geometry file that would create the einzel array. Try to include the proper PA_Define if you can. Seek the assistance of an instructor as necessary.
5. Use New with the Use Geometry files button to access the file manager. Click the edit button to access EDY.
6. Name the file EINZEL.GEM with EDY:
Esc N EINZEL.GEM then **ENTER**
7. Enter the contents into the file from the keyboard.
8. Now Let's save the file, exit EDY, and Select and use this file as the geometry file:
Esc Q Y and you're back in SIMION.
9. Click both buttons on the EINZEL.GEM file's button.
10. If all is OK the array will be created. If not, you will be informed.
11. In any event, enter modify and see what you created.
12. Click the GeomF button to begin to change or fix your geometry definitions

13. If you want to see a successful example, look at the EINZEL1.GEM file.

This page intentionally left blank

Session 10 - User Programs

What is a User Program File

- A user program file is an ASCII file with the **.PRG** filename extension that contains the user program segments for use with a potential array (.PA or .PA0).
- It shares the name of the potential array it supports. **TEST.PRG** would automatically be used to support the **TEST.PA** array.
- User programs are associated with arrays **not** instances.

How SIMION Utilizes User Programs

- Each time the Fly'm button is clicked SIMION automatically compiles all user program files for any instances that have user programs attached to their associated potential array.
- If Adjustable variables are found, the user will be given a screen to view and change their initial values.
- User program segments are called only when ions are flying within the array that the user program segments support.

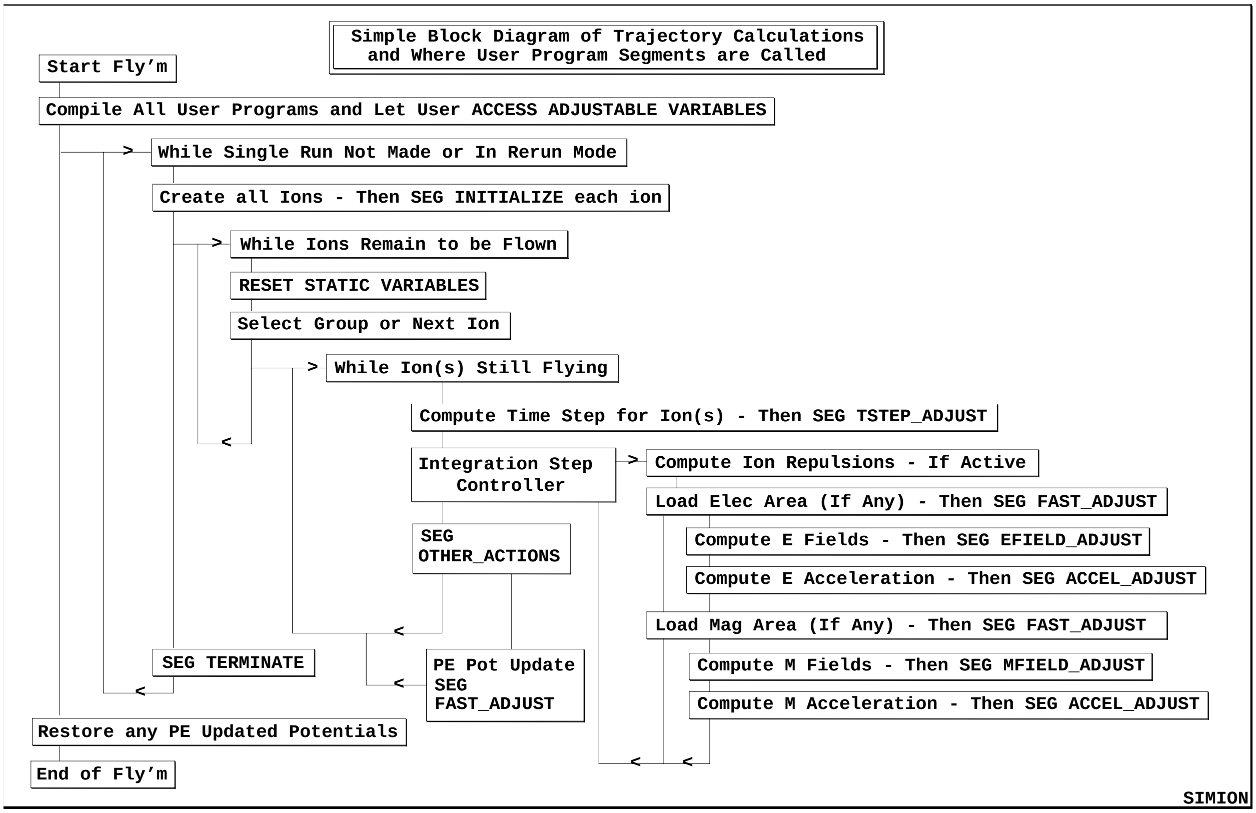
User Program Segments

- The user program file contains user program segments.
- User program segments act like subroutines.
- SIMION calls each type of program segment at specific points in a trajectory calculation.
- Thus you must use a specific user program segment to control a specific aspect of the trajectory calculation.

The Eight Executable Program Segments

1. Program_Segment Initialize
2. Program_Segment Tstep_Adjust
3. Program_Segment Fast_Adjust
4. Program_Segment Efield_Adjust
5. Program_Segment Mfield_Adjust
6. Program_Segment Accel_Adjust
7. Program_Segment Terminate
8. Program_Segment Other_Actions

Where Program Segments are Used


Simple Block Diagram of Trajectory Calculations
and Where User Program Segments are Called
Start Fly'm
Compile All User Programs and Let User ACCESS ADJUSTABLE VARIABLES
While Single Run Not Made or In Rerun Mode
Create all Ions - Then SEG INITIALIZE each ion
While Ions Remain to be Flown
RESET STATIC VARIABLES
Select Group or Next Ion
While Ion(s) Still Flying
Compute Time Step for Ion(s) - Then SEG TSTEP_ADJUST
Integration Step Controller
Compute Ion Repulsions - If Active
Load Elec Area (If Any) - Then SEG FAST_ADJUST
Compute E Fields - Then SEG EFIELD_ADJUST
Compute E Acceleration - Then SEG ACCEL_ADJUST
SEG OTHER_ACTIONS
Load Mag Area (If Any) - Then SEG FAST_ADJUST
Compute M Fields - Then SEG MFIELD_ADJUST
Compute M Acceleration - Then SEG ACCEL_ADJUST
SEG TERMINATE
PE Pot Update SEG FAST_ADJUST
Restore any PE Updated Potentials
End of Fly'm
SIMION

Example of a User Program File

A Questionable Programming Style

```
defa viscous_damping 0,seg accel_adjust rcl ion_ax_mm
rcl ion_vx_mm rcl viscous_damping * - sto ion_ax_mm
rcl ion_ay_mm rcl ion_vy_mm rcl viscous_damping * -
sto ion_ay_mm rcl ion_az_mm rcl ion_vz_mm
rcl viscous_damping * - sto ion_az_mm
```

A Suggested Programming Style

```
; This Accel_Adjust program segment adds a simple Stokes' Law Viscosity Effect
; It serves as a starting point for dabbling in ion mobility and atmospheric ion sources.
; The program segment makes use of a simple viscous damping factor and linear viscosity
; Note:  It has problems with high viscosity - see Accel_Adjust below for fixed version


;Note:  a Begin_Segment Define_Data is not required (the compiler assumes it)

        Define_Adjustable Viscous_Damping 0     ; adjustable variable Viscous_Damping
                                                ; set to 0 (no viscous damping by default)
                                                ; adjustable at the beginning of each Fly'm


Begin_Segment Accel_Adjust                      ; start of Accel_Adjust program segment


        Recall Ion_Ax_mm                        ; recall current x acceleration (mm/usec^2)
        Recall Ion_Vx_mm                        ; recall current x velocity (mm/sec)
        Recall Viscous_Damping                  ; recall the viscous damping term
        Multiply                                ; multiply times x velocity
        Subtract                                ; and subtract from x acceleration
        Store Ion_Ax_mm                         ; return adjusted value to SIMION

        Recall Ion_Ay_mm                        ; recall current y acceleration (mm/usec^2)
        Recall Ion_Vy_mm                        ; recall current y velocity (mm/sec)
        Recall Viscous_Damping                  ; recall the viscous damping term
        Multiply                                ; multiply times y velocity
        Subtract                                ; and subtract from y acceleration
        Store Ion_Ay_mm                         ; return adjusted value to SIMION

        Recall Ion_Az_mm                        ; recall current z acceleration (mm/usec^2)
        Recall Ion_Vz_mm                        ; recall current z velocity (mm/sec)
        Recall Viscous_Damping                  ; recall the viscous damping term
        Multiply                                ; multiply times z velocity
        Subtract                                ; and subtract from z acceleration
        Store Ion_Az_mm                         ; return adjusted value to SIMION

Exit                                            ; exit to SIMION (optional statement)
```

Language Rules

- Case is Ignored (upper and lower case are the same)
- Blank lines and indentions are ignored
- The Semicolon ; Starts an In-Line Comment
- The language is based on the HP RPN calculator format with a 10 register rotary stack:

  ```
  25 10 * 5 /        ;  means (25 * 10) / 5
  ```

- The Compiler looks for words and converts them into
  Commands
  Numbers
  Variable Names and Labels

Examples of Commands

+ or: Add

Adds contents of x and y registers, puts result in y-register, and renames it as x-register (*e.g.* **1 2 +** *becomes 3 in register where 1 was originally stored*).

>KE or: Speed_to_Kinetic_Energy

Converts from speed (*mm/usec*) to kinetic energy (*eV*). On entry the x-register is assumed to contain the ion's speed and the y-register is assumed to contain the mass of the ion (*amu*). On exit the x-register contains the ion's KE and the y-register is unchanged. ***This transform uses relativistic corrections***. The **>SPD** command performs the reverse transformation.

Classes of Commands

- Calculations
  + - * / SIN COS LOG

- Transformations
  >DEG >RAD >KE >P3D >WBC

- Flow Controls
  LBL GSB RTN EXIT

- Tests (do next command if true else skip next command)
  X=0 X<=0 X!=Y X>Y

- Communication to User and Variables
  STO RCL R/S KEY? MESSAGE

Types of Variables

- Adjustable Variables (defined at top of program file)
  DEFA MY_VARIABLE 30.0
  Global and lasts ***throughout*** a Fly'm

- Static Variables (defined at top of program file)
  DEFS TIME_FLAG -2.0E4
  Global and Reset to initial value ***before*** each ion is flown

- Temporary Variables
  STO LUCKY
  Local and Temporary: Created with STO command. Name must be ***unique*** (not Adjustable, Static, or Reserved variable name)

- Reserved Variables
  Ion_Color Ion_Charge Ion_Px_mm
  Reserved variables are use to communicate with SIMION. Each Program Segment has specific read and write access to various reserved variables.

Name	Use	Units	Read Access	Write Access
Adj_Elect01 to Adj_Elect30	Fast Adj Electrode Voltages	Volts	Fast_Adjust	Fast_Adjust
Adj_Pole01 to Adj_Pole30	Fast Adj Pole Mag Potentials	Mags	Fast_Adjust	Fast_Adjust
Ion_Ax_mm Ion_Ay_mm Ion_Az_mm	Ion's current Acceleration (WB Coordinates)	mm/micro sec^2 (WB Orientation)	Accel_Adjust Other_Actions Terminate	Accel_Adjust
Ion_BfieldX_gu Ion_BfieldY_gu Ion_BfieldZ_gu	Magnetic Field at Ion's Location (PA's Orientation)	Gauss (PA's Orientation)	Mfield_Adjust	Mfield_Adjust
Ion_BfieldX_mm Ion_BfieldY_mm Ion_BfieldZ_mm	Magnetic Field at Ion's Location (WB Orientation)	Gauss (WB Orientation)	Accel_Adjust Other_Actions Terminate	None
Ion_Charge	Ion's current charge	in units of elementary charge	Any Prog Seg	Initialize Other_Actions
Ion_Color	Color of Ion	0-15	Initialize Other_Actions	Initialize Other_Actions
Ion_DvoltsX_gu Ion_DvoltsY_gu Ion_DvoltsZ_gu	Voltage Gradient at Ion's Location (PA's Orientation)	Volts/grid unit (PA's Orientation)	Efield_Adjust	Efield_Adjust
Ion_DvoltsX_mm Ion_DvoltsY_mm Ion_DvoltsZ_mm	Voltage Gradient at Ion's Location (WB Orientation)	Volts/mm (WB Orientation)	Mfield_Adjust Accel_Adjust Other_Actions Terminate	None
Ion_Instance	Current Instance	1- max instance	Any Prog Seg	None
Ion_Mass	Ion's current mass	amu	Any Prog Seg	Initialize Other_Actions
Ion_mm_Per_Grid_Unit	Min Current Scaling[1]	mm/grid unit	Any Prog Seg	None
Ion_Number	Ion's Number	1- max ion	Any Prog Seg	None
Ion_Px_Abs_gu Ion_Py_Abs_gu Ion_Pz_Abs_gu	Ion's current PA Array Coordinates	grid units (PA's Abs Coordinates)	Any Prog Seg	None
Ion_Px_gu Ion_Py_gu Ion_Pz_gu	Ion's current (PA's Coordinates)	grid units (PA's Coordinates)	Any Prog Seg	None
Ion_Px_mm Ion_Py_mm Ion_Pz_mm	Ion's current Workbench Coordinates	mm (WB Coordinates)	Any Prog Seg	Initialize Other_Actions
Ion_Splat	Ion Status Flag[2]	Flying = 0 Not Flying != 0	Initialize Other_Actions	Initialize Other_Actions
Ion_Time_of_Birth	Ion's Birth Time	micro seconds	Any Prog Seg	Initialize Other_Actions
Ion_Time_of_Flight	Ion's current TOF[3]	micro seconds	Any Prog Seg	Other_Actions
Ion_Time_Step	Ion's Time Step[4]	micro seconds	All but Initialize	Tstep_Adjust
Ion_Volts	Electrostatic Pot at Ion's Location	Volts	Efield_Adjust Mfield_Adjust Accel_Adjust Other_Actions Terminate	Efield_Adjust
Ion_Vx_mm Ion_Vy_mm Ion_Vz_mm	Ion's current Velocity (WB Coordinates)	mm/micro sec (WB Orientation)	Any Prog Seg	Initialize Other_Actions

Rerun_Flym	Rerun Flym Flag	NO = 0
		YES = 1
			Initialize
			Terminate
				Initialize
				Terminate

Update_PE_Surface	New PE Surface[5]	YES != 0	None	Other_Actions

Session 10 - Lab on User Programs

This Lab's Files are found in Directory: C:\CLASS\SESSON10

Using an Existing User Program

The best way to start programming is to steal, steal, steal.

1. Remove all PAs From RAM
2. Use New to create a new array using EINZEL.GEM from directory C:\CLASS\SESSON10
3. Save the array as EINZEL.PA# and ***then*** Refine it.
4. Remove all PAs From RAM
5. Use View to load EINZEL.IOB
6. Use the Define button and load the EINZEL.FLY file.
7. Fly the ions and see that they focus properly.

8. Now exit View and click on the GUI file manager button
9. Click the Other button and enter the following command in the DOS ioline:
   COPY RANDOM.PRG EINZEL.PRG
   and press the **ENTER** key (to create a copy of the user program with the EINZEL name).
10. Hit the Esc key several times to return to the main menu.
11. Notice that the Test & Debug User program button is unblocked. SIMION has recognized that there is now a user program associated with the EINZEL.PA0 potential array. Click the Test & Debug button access the user program development system (just a quick look).
12. Click the Compile button and the program will be compiled. If you want a cross reference listing depress the Show Xref Listing button and then click the Compile button again. If you want to look at this cross reference listing with EDY click the View: EINZEL.ERR button. Look at the listing. You can exit EDY by the following keystrokes: **ESC Q N**.
13. To examine the user program file with EDY click the Edit: EINZEL.PRG button. Look at the listing. Exit EDY by the following keystrokes: **ESC Q N**.
14. Click the Compile button again to verify everything is OK. Now click the Quit key to return to the main menu.
15. Re-enter View and fly the ions. SIMION will compile the user programs and display the adjustable variables. Click the E Fmt button to switch from E to F format display. Now fly the ions with the default settings. What a mess!
16. You can turn off the user programs for a single run. Click Fly'm and then Click the On button to turn the user program off, and then fly the ions.
17. Re-Fly the ions with various adjustable variable settings and see what happens. Are the results reasonable?
18. Perhaps you would like to change the starting values for the adjustable variables in the user program. Click the PAs tab and then the Debug button to access the user program development screen. Now click the Edit button to access the user program file with EDY. Change the starting values of the adjustable variables. Now save the file and exit from EDY: **ESC Q Y**. Click the Compile button to check for compile errors. Now click the Quit button to return to View. Re-Fly the ions to confirm that the changes in initial value have been made successfully

Adding a User Program Segment

User programs generally develop by evolution. In this can we would like to exert some user control from the keyboard. The object is to create a marker every time a keyboard key is pressed. This is really a very simple task.

1. Click the PAs tab and then the Debug button to access the user program development screen. Now click the Edit button to access the user program file with EDY.
2. Add the following two lines at the bottom of the file:

```
seg other_actions
        key? x!=0 mark
```

   The first line defines the start of an Other_Actions program segment. The second line has three commands:

   key? Tests for keyboard input and returns a non-zero key code if you hit a key

   x!=0 Tests the return value from key? for a non-zero value (you hit a key)
   If the value is non-zero the next instruction will be executed (do if true) else it will be skipped and the program segment will exit without doing anything.

   mark This command creates a single marker. It uses the currently selected marker color and can be used as a trigger event for data recording. The mark command is executed if the x!=0 test is true.

3. Now save the changes and exit EDY: ESC Q Y. Test compile your efforts. If you have an error fix it with EDY. Keep trying until it compiles OK. Now click the Quit button to return to View.
4. Fly the ions as dots with the speed slider to the center to slow the ions down a bit. Hit the spacebar and watch for markers. Can you create markers anywhere? What is the problem? ***Remember, an array's user program will only be called when ions are actually within its instance's array volume***.

Additional Challenges

1. Modify the user program to output the ion's x, y, and z position in workbench coordinates (mm) when you hit a key. Hint: The **MESSAGE**, **LBL**, **GTO**, and **EXIT** commands might be helpful.
2. Change this program to output x, y, and z in 3D array coordinates and then in absolute array coordinates.
3. For a real challenge calculate and display the ion's KE when you hit a key. Hint: The **>P3D** and **>KE** transforms might be helpful.