# Realistic Simulation of the Ion Cyclotron Resonance Mass Spectrometer Using a Distributed Three-Dimensional Particle-In-Cell Code

Dale W. Mitchell
Department of Physics and Astronomy, Ohio University, Athens, Ohio, USA

This work describes an Internet accessible three-dimensional particle-in-cell simulation code, which is capable of near first principles modeling of complete experimental sequences in Fourier transform ion cyclotron resonance mass spectrometers. The graphical user interface is a Java client that communicates via a socket stream connection over the Internet to the computational engine, a server that executes the simulation and sends real-time particle data back to the client for display. As a first demonstration, this code is applied to the problem of the cyclotron motion of two very close mass to charge ratios at high ion density. The ion populations in these simulations range from 50,000 to 350,000 coulombically interacting particles confined in a cubic trap, which are followed for 100,000 time-steps. Image charge, coherent cyclotron positions, and snapshots of the ion population are recorded at selected time-steps. At each time-step in the simulation the potential (coulomb + image + trap) is found by the direct solution of Poisson's equation on a $64 \times 64 \times 64$ computational grid. Cyclotron phase locking is demonstrated at high number density. Simulations at different magnetic fields confirm a $B^2$ dependence for the minimum number density required to lock cyclotron modes. (J Am Soc Mass Spectrom 1999, 10, 136–152) © 1999 American Society for Mass Spectrometry

A realistic three-dimensional (3D) many particle simulation of a trapped ion mass spectrometer should incorporate space charge effects (coulomb and image charge interactions) for a sufficiently large ion population, and correctly model all major aspects of the experiment including trap boundary, applied potentials, magnetic field, and neutral bath gas. The purpose of this work is to describe the development of such a three-dimensional many particle simulator and to give a preliminary demonstration of this code by applying it to the problem of cyclotron phase locking of two closely spaced masses at high space charge conditions [1–8].

A Fourier transform ion cyclotron resonance mass spectrometer (FTICR) consists of a strong magnetic field, which is directed parallel to the symmetry axis ($z$ axis) of the ICR trap [9–13]. The magnetic field confines the ions radially while the trap [14–17] creates a potential well to confine the ions longitudinally. An ICR weighs ions by measuring their cyclotron frequencies. The simplest possible FTICR experiment consists of successive steps of ion accumulation, cyclotron mode excitation and image charge detection. Applying an excitation at the ions' cyclotron frequencies coherently excites the cyclotron modes. The ions are subsequently followed for a long detection period by monitoring the image charge induced on selected trap electrodes. A Fourier transform of this signal yields the individual cyclotron frequencies.

Considerable progress has been made towards understanding how various pieces of an ICR experiment perturb the three-dimensional motion of a single ion [18]. In general there are many simultaneously confined ions with one or more different mass/charge species. Several models have been proposed [7, 19–23] to account for space charge induced cyclotron frequency shifts, usually based on determining an average radial force arising from an assumed charge distribution which neither changes its shape nor its cyclotron radius during detection. These simple models predict the correct magnitude of the experimentally observed downward shift under conditions that the average force model is appropriate, namely relatively low number density and wide cyclotron frequency separation between different species. As discussed below, at high number density, low magnetic field, or closely spaced cyclotron frequency separation the dynamics becomes considerably more complicated than just a frequency shift. One method to theoretically investigate the high

Address reprint requests to Dr. Dale Mitchell, Department of Physics and Astronomy, Ohio University, Athens, OH 45701. E-mail: mitchell@helios.phy.ohiou.edu


1044-0305/99/$20.00
PII S1044-0305(98)00130-5

Received June 22, 1998
Revised October 8, 1998
Accepted October 12, 1998

space charge regime in FTICR is by a realistic computer simulation.

Most simulation work to date in FTICR and ion trap mass spectrometry has focused on studying the effects of ion–neutral collisions or various external fields on single particle trajectories [23–29]. SIMION is an example of a popular general-purpose code for studying problems of this type [30]. In contrast few simulation studies have been published on large populations of coulombically interacting particles in ICR or ion trap mass spectrometers. Simple two-dimensional simulation programs have previously been used to investigate varied phenomena related to coulomb interactions in FTICR [7, 8, 31–33]. Three-dimensional FTICR simulations involving a relatively small number of interacting particles have also been carried out [8, 34–36]. Miluchihin, Miura and Inoue developed a 3D parallel code to integrate the equations of motion of up to 1024 interacting particles for 50,000 time-steps using a 1024 processor computer [34].

Recently [37], two-dimensional many particle simulations of trapped ion mass spectrometers in Cartesian and cylindrical confinement geometries have been carried out utilizing the particle-in-cell algorithm with Monte Carlo ion–neutral collisions [38–40]. The particle-in-cell method (PIC) is a computational algorithm for efficiently solving the dynamics of very large populations of coulombically interacting particles [38–40]. The computational effort increases approximately linearly in the number of interacting particles. Linear scaling is achieved by solving Poisson's equation on a computational grid. Poisson's equation is solved at each simulation time-step with boundary conditions and charge density appropriate for the particular electrode geometry and ion population, respectively. Field quantities are evaluated on the grid and interpolated to the particles.

The primary advantage of the two-dimensional PIC simulations is that the solution to Poisson's equation, which is calculated at every time-step, can be done much faster than in three dimensions. Another advantage of two-dimensional results is that the number of independent variables is reduced compared to three-dimensional allowing for a more thorough study of the accessible parameter space. The principle criticism of two-dimensional results is that the experimental apparatus and the ion trajectories are three-dimensional which means that two-dimensional results must be supported by other information such as insight and experimental data to justify the two-dimensionality of the problem. On the other hand, an accurate fully 3D simulation code, incorporating all major processes in a first principles manner, should lead to near unambiguous, quantitative results.

The present work describes a new three-dimensional particle-in-cell code that is capable of modeling a complete FTICR experiment at high space charge conditions. This code, called PIC3D, is applied to ion populations of up to 350,000 coulombically interacting particles for 100,000 time-steps, while maintaining correct trap boundary conditions. These are the first realistic three-dimensional ICR simulations involving very large ion populations. This code is sufficiently general to handle almost any conceivable simulation sequence and experimental parameters. After a short review of the particle-in-cell algorithm, a description of the PIC3D client-server program is presented. As a first application, PIC3D is applied to cyclotron phase locking between two close masses at sufficiently high ion density.

## Particle-in-cell Algorithm

The particle-in-cell algorithm [37–40] follows several basic steps at each time-step in the simulation. Charge density is interpolated from the continuous particle positions to the discrete grid points on the computational grid by using volume weighting, which is the three-dimensional extension of area weighting described in earlier work [37–40]. Using this charge density with the correct trap boundary conditions, the potential is obtained on the computational grid using a direct Poisson solver. The electric fields at the particle positions are calculated by interpolation from the grid electric fields using volume weighting. The particle positions and velocities are advanced in time for one time-step by the second-order leap-frog integrator. Elastic ion-neutral collisions are implemented by the Monte Carlo method. Any particles, which have left the confinement geometry, are removed. This procedure is repeated for the next time-step.

The potential, including space charge is determined by a direct solution of the three-dimensional Poisson equation at each simulation time-step. Since the potential is solved at each time-step and the trap boundaries coincide with the boundary conditions used in solving Poisson's equation, the potential is also correct at the trap walls which allow a first-principles evaluation of the image charge and, hence, the ICR detected signal. The 3D PIC algorithm used in the present work is the three-dimensional extension of a two-dimensional Cartesian PIC described in detail elsewhere [37]. A few of the important components of the PIC3D computational engine are outlined below.

### *Three-dimensional Poisson Solver*

The interior of the tetragonal ICR trap is divided into a uniform computational grid. The finite difference form of Poisson's equation is solved on this grid using a direct 3D Poisson solver based on the three dimensional fast Fourier transform (3D-FFT) solution of Poisson's equation [38, 39] with Dirichlet boundary conditions. The charge density used in Poisson's equation is calculated by interpolating charge from the continuous particle positions to the eight nearest neighbor grid points using a volume weighting scheme. The potential obtained in this manner contains self-consistent contributions from the applied trap potentials, image charge

induced on the electrodes and the coulomb interaction between charged particles. The resulting potential is correct everywhere within the trap including at the trap walls. Interpolating charge at the continuous particle positions to the discrete grid points on which Poisson's equation is solved implicitly replaces point particles with finite sized particles (diameter on the order of the distance between neighboring grid points). The electric fields at the particle positions are interpolated from the potential at the discrete grid points using the same volume weighting scheme as used for the charge density. In the current implementation the user specifies applied potentials to the six trap plates. Any conceived static or time-dependent potential can be applied. The code described below is applied to a system of $>10^5$ coulombically interacting particles for $10^5$ time-steps using a $64 \times 64 \times 64$ computational mesh.

The direct Poisson solver is very efficient. For example, a 500 MHz Dec/Alpha workstation running the Linux operating system solves Poisson's equation to double precision on $32^3$, $64^3$, and $128^3$ grids in 0.06, 0.64, and 6.8 s, respectively. Each factor of 8 in the number of grid points increases the time for solving the potential by an order of magnitude. While this FFT Poisson solver is most efficient if the number of grid points is a power of 2, this is not a restriction in the code. For example, a $100^3$ grid requires 2.7 s of CPU time. Also, the number of grid points in each dimension can be different (e.g., a $64 \times 64 \times 128$ grid can be used to study an elongated trap whose length is two times the trap width). It is also possible, though not done in the current implementation, to use the direct Poisson solver for any arrangement of trap electrodes, and hence any complex trap geometry or arrangement of interior trap electrodes, by use of the capacitance matrix method [37, 39]. The capacitance matrix method requires solving Poisson's equation twice per time-step.

### *Particle Push*

The particles' positions and velocities are advanced forward in time for one time-step by the second-order explicit leap-frog algorithm [36–40]. The leap-frog integrator is absolutely stable when the product of the time-step and the fastest angular frequency present in the system is less than two.

### *Ion–Neutral Collisions*

When a neutral bath gas is present, ion–neutral elastic collisions are implemented by the Monte Carlo algorithm [28, 34, 37, 38]. Given the neutral gas mass, temperature, pressure, neutral polarizability, and/or hard sphere cross section, ion–neutral Langevin and/or hard sphere collisions are accurately treated by this method. At the beginning of a simulation run a random sample of about $10^4$ neutral collision gas speeds are taken from the Maxwell-Boltzmann distribution and stored in an array. In addition an array of about $10^5$ points on a unit radius sphere (unit vectors) are accumulated. At each time-step for each particle in the simulation, a neutral gas velocity vector is chosen by randomly selecting a speed from the Maxwell-Boltzmann distribution and multiplying this speed by a random unit vector. The probability of a collision is calculated and compared with a random number to determine whether an elastic collision occurs. If a collision occurs, the particle's center of mass velocity is given a random direction (isotropic scattering) and then transformed back to the lab frame.

## PIC3D: A Distributed 3D Particle-in-cell Code

PIC3D is an Internet accessible three-dimensional particle-in-cell code developed by the author. The graphical user interface (client) and computational engine (server) reside on different computers and communicate over the Internet using a socket stream connection.

The graphical user interface is written entirely in Java (v1.1) allowing the client to run on any computing platform or appliance that the Java virtual machine has been ported. Java is an object oriented programming language developed by Sun Microsystems which has powerful network programming capabilities and allows for cross-platform software development. Since the graphical user interface is the most platform specific part of almost any program, writing the user interface entirely in Java has the advantage of portability across most platforms. For example, the PIC3D client has been ported to Win95/NT, Intel Linux, Dec/Alpha Linux, and Sun Sparc Solaris. These are the only platforms tested. The PIC3D client should be able to run on any Java supported platform with minimal, if any, code modification.

The most CPU and memory critical part of PIC3D is the portion of the code that computes the particle-in-cell algorithm. The particle-in-cell algorithm is computed by the server, which resides on a different computer than the Java client. The PIC3D server is written entirely in C and communicates with the remote Java client over the Internet using a socket stream connection (TCP/IP sockets). The PIC3D server has been compiled with the GNU C compiler (gcc) on Intel Linux, Dec/Alpha Linux, and Sun Sparc Solaris platforms. The basic portability requirement for the server is that the platform has a sockets library. The server runs in the background, listening for requests from a remote client. If a request is made, the server forks a new process depending on the type of client making the request. If the request is from the Java client, the server initiates a new many particle simulation. If an HTTP client (web browser) makes a request to the IP address and port that the PIC3D server listens, then the PIC3D server emulates a web server. The server creates an ensemble snapshot of an on-going simulation and sends these snapshots, GIF images created on the fly, back to the

requesting web browser. In this manner the real-time status of a simulation in progress, perhaps requiring several days of CPU time, can be remotely queried at any time from a web browser.

A major advantage gained in separating the computational engine, which executes the particle-in-cell algorithm, from the user interface is that the server can communicate with any privileged remote client. This client-server architecture provides an efficient means to incorporate future code extensions as computing technology evolves without having to remain rooted in one specific platform. Hence, PIC3D is highly extensible, modular, and portable without having to sacrifice the performance of the computational engine.

Two versions of the Java client have been developed; a standalone application and an applet. An applet requires a Java enabled web browser to run while the standalone application runs independent of a web browser. The standalone application is more efficient and powerful compared to the applet version of the client. Among the advantages of the standalone application are its ability to run simulations on several different computers, simultaneously, and its ability to access local system resources. The standalone application is also much faster in both execution speed and in reading data over the network, possibly since the Java bytecodes is compiled using a high quality just-in-time compiler for the standalone and does not have the overhead of a web browser to deal with.

A user initiates a simulation by opening a number of windows from the PIC3D Java client including windows for specifying ICR simulation parameters, initial conditions, displayed diagnostics and network information. Once the simulation begins running, the user views various diagnostics such as particle positions for a representative sample population in real-time and the complete trajectories of a few particles. Figure 1 is a screen shot of the primary PIC3D client window of a simulation in progress. This window displays real-time particle data for a subset of the total ion population including snapshots of the current particle positions in $x$–$y$ (upper left in Figure 1) and $y$–$z$ (lower left) perspective. Also displayed is the complete 3D trajectory of one ion inside the cubic trap. This simulation consisted of four sequence steps: ion accumulation, collisional cooling, cyclotron excitation, and detection. The 3D box can be rotated and resized by the mouse allowing for a close inspection of individual ion trajectories. At user specified time intervals various diagnostics are stored on the server including snapshots (GIF images) of the particle ensemble, image charge induced on selected electrodes, and coherent cyclotron positions for each species. In addition, the energy (total, kinetic, electrostatic), canonical angular momentum about the $z$ axis (total, magnetic, mechanical), and number of simulation particles are recorded.

The current implementation of PIC3D permits the user to set up multiple sequence simulations similar to an actual laboratory experiment. The user opens the sequence window and fills out parameters for a simulation that may require up to five different simulation steps. For example, separate sequence steps consisting of ion accumulation, collisional cooling, dipolar excitation and detection can be specified in one simulation run. The time-step can be different for each sequence step. The user interface is sufficiently flexible to allow any combination of static or time-dependent potential on any of the six trap walls. A single frequency or frequency sweep excitation can be applied to any trap wall. In addition the client and server can easily be extended to accommodate more complicated trap and excitation schemes such as various segmented electrode configurations.

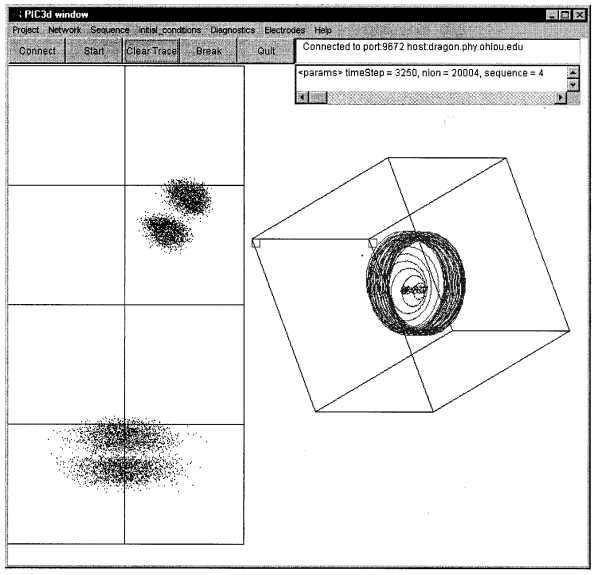


**Figure 1.** Screenshot of the main PIC3D Java client window. At user specified time intervals the PIC3D server sends selected particle position data back to the client for real-time display. Particle positions at the current time-step are plotted in $xy$ perspective (upper left) and $zy$ perspective (lower left). The complete three-dimensional trajectory (right) of one particle is plotted inside of the cubic trap, which can be rotated with the mouse. At the beginning of a simulation the user opens various windows to specify simulation conditions using the menu bar. Clicking on the "Connect" button initiates an Internet connection with the server. The "Start" button causes the client to send simulation parameters to the server. The server then begins the simulation. The "Break" button disconnects client-server communication with the simulation continuing to run on the server. The "Quit" button disconnects client-server communication and ends the on-going simulation.

The remote user can remain connected to the PIC3D server for the complete simulation. However, there are situations when a user would like to run a simulation without remaining connected to the server. These cases may include a simulation requiring several hours or days of CPU time, freeing up resources for some other application, or to use the PIC3D client to initiate a second simulation on a different machine. The client can disconnect from the server without stopping execution of the simulation. Once this occurs, in the present imple-

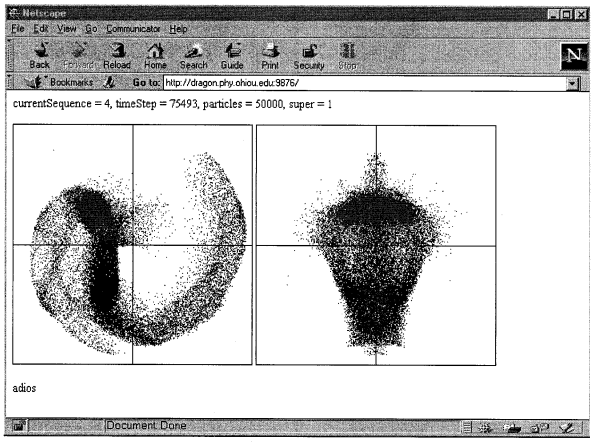


**Figure 2.** Real-time status of a simulation via an HTTP client. A popular web browser is used to query the PIC3D server on the status of an on-going simulation. The server responds by forking a new process that creates GIF images of the current time-step simulation, and sends this data back to the requesting web browser. The particle positions are plotted in *xy* (left) and *zy* (right) perspective.

mentation the user cannot connect back to the server using the PIC3D client. However, the status of the simulation at any time is viewable by a web browser, even when the PIC3D server host computer does not have a web server. This is because the PIC3D server listens on a specified port for incoming requests. These requests can be any client, which specifies the IP address of the host computer and the PIC3D server port address, including the PIC3D Java client or an ordinary web browser. The PIC3D server recognizes different types of clients and takes appropriate response. Direct requests from a web browser cause the server to create a snapshot (GIF images are created on the fly) of the current time-step ion population and to send these images back to the requesting web browser. Figure 2 shows a screen shot of using a web browser to inquire the status of an on-going simulation. A request from an HTTP client to port 9876 (the port which the PIC3D server is listening in Figure 2) causes the PIC3D server to create a snapshot of an on-going simulation.

The simulations described below were carried out using the Java standalone application for the PIC3D client on a Win95 PC platform with a 166 MHz Intel CPU and 32 MB RAM. The PIC3D server was executed on a Linux platform with a 500 MHz Dec/Alpha CPU and 128 MB RAM. A single simulation run with 50,000 interacting particles, 100,000 time-steps and a $64^3$ computational grid for Poisson's equation required 1 day of CPU time. A 350,000 particle, 100,000 time-step, $64^3$ grid simulation required 4 days of CPU time.

## Application to Cyclotron Phase Locking at High Space Charge Conditions

Below, PIC3D is applied to the computationally intensive problem of following the dynamics of up to several hundred thousand coulombically interacting ions confined in a FTICR mass spectrometer. In this series of simulations two close masses 100.0 and 100.3 u are used. These species are given equal relative abundance and are singly charged. A four-step sequence is used in the simulations with various parameters listed in Table 1. To summarize a cubic ICR trap (width 2.5 cm) is in a uniform magnetic field (1 T) directed parallel to the *z* axis. The unperturbed cyclotron frequencies are close to 153 kHz and the cyclotron frequency difference between the two different masses is about 459 Hz. This indicates that if these two masses have a substantial coherent cyclotron radius that they should pass each other every 2.2 ms. At sufficiently high number density the ion cloud, after cyclotron mode excitation, may not separate completely into two different mass ion clouds, each with a different cyclotron frequency, but rather remain locked together. Several different research groups have observed experimentally [1–5] and studied theoretically [2, 6–8] this phenomenon called cyclotron phase locking (or peak coalescence).

**Table 1.** Parameters for simulations at 1 T

| Parameter | |
|---|---|
| Trap potential | 1.0 V |
| Magnetic field | 1.0 T |
| Cubic trap width | 2.5 cm |
| Initial cloud radius | 1.0 or 1.5 mm |
| Initial cloud length | 2.2 cm |
| Grid size | $64 \times 64 \times 64$ |
| Number of simulation particles | 50,000–350,000 |
| Ion masses | 100.0 and 100.3 u |
| Ion charge | 1 e |
| Super (number of ions per simulation particle) | 1 |
| 1. Accumulation sequence duration | 0.7 ms |
| Accumulation time-step | 0.70 $\mu$s |
| Number of ions injected per time-step | 50–350 |
| Number of accumulation sequence time-steps | 1000 |
| 2. Collision sequence duration | 2.0 ms |
| Collision time-step | 0.70 $\mu$s |
| He neutral mass, pressure, temperature | 4 u, 1.5 mtorr, 300 K |
| Number of collision sequence time-steps | 2800 |
| 3. Excitation sequence duration | 44.6 $\mu$s |
| Excitation time-step | 0.030 $\mu$s |
| Excitation potential ($V_{pp}$) | 10.0 V |
| Excitation frequency | 153.4 kHz |
| Number of excitation sequence time-steps | 1488 |
| 4. Detection sequence duration | 6.5 ms |
| Detection time-step | 0.065 $\mu$s |
| Number of detect sequence time-steps | 100,000 |

The cubic trap [14–17] has static potentials of 1 V on the $z$ plates and ground on the plates parallel to the magnetic field. During the cyclotron excitation simulation step, two opposing trap electrodes ($y$ plates) have applied rf potentials with opposite phase set up for single frequency dipolar excitation. The Poisson equation boundary conditions are exact for the cubic trap with applied static and excitation potentials. The simulation sequence follows successive stages of ion accumulation, collisional cooling, dipolar cyclotron excitation then image charge detection. The individual steps are described in detail below.

The first two sequence steps are to load the ions into the trap then to create a thermal equilibrium. In step 1 ions were injected into the trap for 1000 time-steps at a constant current at random positions within an on-axis cylindrical volume of length 2.2 cm (88% of the trap length) and radius of either 1.0 or 1.5 mm. This step results in a nonthermal distribution but with a density whose symmetry axis coincides with the trap $z$ axis. The second stage of the simulation, step 2 in Table 1, creates a thermal equilibrium (300 K) ion population. This is accomplished by allowing the ions to equilibrate with a 300 K Maxwell-Boltzmann distribution He bath gas at 1.5 mtorr pressure for 2 ms through elastic ion–neutral collisions.

Three different simulation runs are described and compared in detail. These represent low, medium, and high number density simulations, each with unique dynamical behavior. The total ion populations are $50 \times 10^3$, $150 \times 10^3$, and $350 \times 10^3$ numbers of ions. These are the numbers of particles used in the simulations (each simulation particle corresponds to exactly one ion). For simplicity, the different simulations will be referred to as 50k, 150k, and 350k, respectively. Two different initial cloud radii are used. The initial cloud radius is a simulation parameter that gives the radius of the cylindrical volume used to inject ions during the ion accumulation step of the simulation, step 1 in Table 1. The 50k, 150k, and 350k simulations have initial cylindrical radii of 1.5, 1.0 and 1.0 mm, respectively. Ion–neutral collisions with He reduce the $z$ amplitudes and expand the cloud radially. After ion–neutral collisions, the ions are in a 300 K thermal distribution. Radial and axial density profiles are shown in Figure 3 immediately after the collisional cooling event. The distribution aspect ratios (distribution length divided by diameter) are in the range of 2.5–3.5 with the largest aspect ratio occurring for the largest ion population simulation. The central number densities after collisional cooling for the 50k, 150k, and 350k simulations are $1.1 \times 10^{12}$, $4.9 \times 10^{12}$, and $6.8 \times 10^{12}$ m$^{-3}$ with an uncertainty of $\pm 0.2 \times 10^{12}$ m$^{-3}$. The density limit for an ion cloud consisting of 100 u ions in a 1 T magnetic field is $2.66 \times 10^{13}$ m$^{-3}$, the Brillouin limit [41, 42] $n_B = \varepsilon_0 B^2/2m$. Therefore, the low, medium, and high ion populations used in the simulations correspond to distributions with central densities that are 4%, 18%, and 26% of the trap density limit before cyclotron mode excitation.

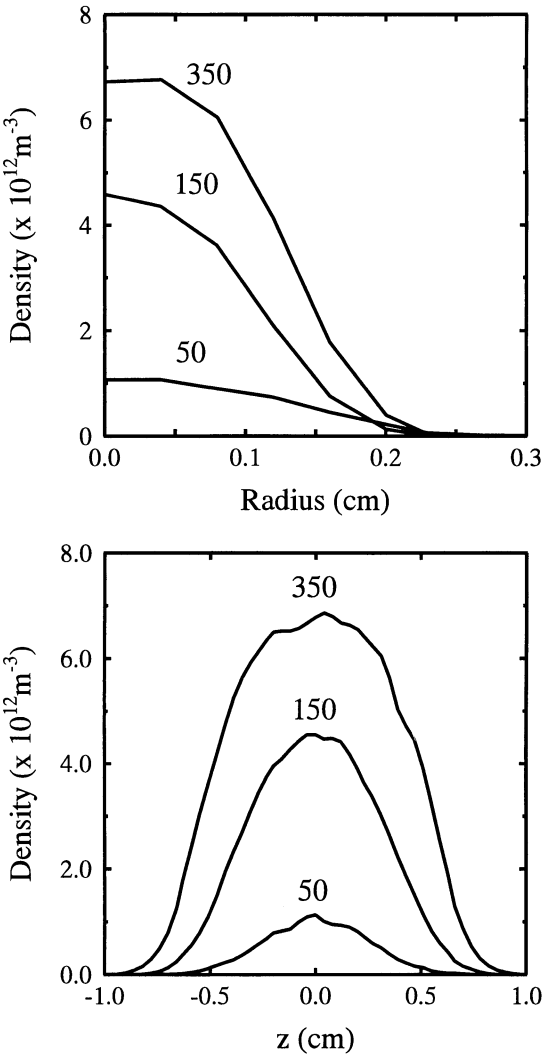


**Figure 3.** Density profiles immediately after the ion-neutral collision sequence, in the radial direction (top) and along the $z$ axis (bottom).

Step 3 is the cyclotron mode excitation sequence. The cyclotron modes of the ions are excited by a single frequency dipolar excitation ($V_{pp} = 10.0$ V) applied at a frequency of 153.4 kHz for 44.6 $\mu$s. Using the familiar relation for the coherent cyclotron radius of a single ion initially at the trap center and excited by a linear excitation ($R_c = ET_{ex}/2B \cong 0.72V_{pp}T_{ex}/2Bd$, for a cubic trap) [16], the predicted coherent cyclotron radius is 0.64 cm for a single ion. It turns out that this model prediction is surprisingly close (within 0.02 cm) to the coherent cyclotron radii obtained from the PIC3D simulation for all initial distributions using these excitation parameters. It should be remembered that the simulations involve large numbers of coulombically interacting ions, which have three-dimensional spatial distributions and that the excitation used in the simulations is exact for the cubic trap. On the other hand, the simple calculation is for a single ion initially at the trap center and excited by a linear excitation.

The detection phase of the simulation is step 4. The ions are followed for 100,000 time-steps, corresponding to 6.5 ms detection time. This time is sufficient to elucidate whether cyclotron phase locking occurs since the cyclotron frequency difference between the two different species of about 459 Hz corresponds to a cyclotron beat period of 2.2 ms. Neglecting coulomb interactions two different mass species with a substantial coherent cyclotron mode separate into two different ion clouds with different cyclotron frequency.

If the two ion clouds are locked together, indicating the same detected cyclotron frequency, there are clear signatures in the simulation results to this effect. The most reliable method to detect cyclotron phase locking is probably to view a movie of the ion ensemble over the complete simulation. The signature of a phase locked pair of ion clouds with different mass to charge ratios is that the two clouds will not separate into two different ion clouds with different cyclotron frequency [2, 6–8]. A second direct method to detect cyclotron phase locking in two coherent ion clouds which have coherent cyclotron radii larger than their cloud radii is to calculate the separation distance between their coherent cyclotron positions. The separation distance for an unlocked pair varies with time from close to zero to a maximum separation of twice the cyclotron radius. A phase locked pair of ion clouds will have a separation distance less than the cyclotron radius. A third, less reliable, signature for cyclotron phase locking is to Fourier transform the detected signal to obtain the frequency spectrum.

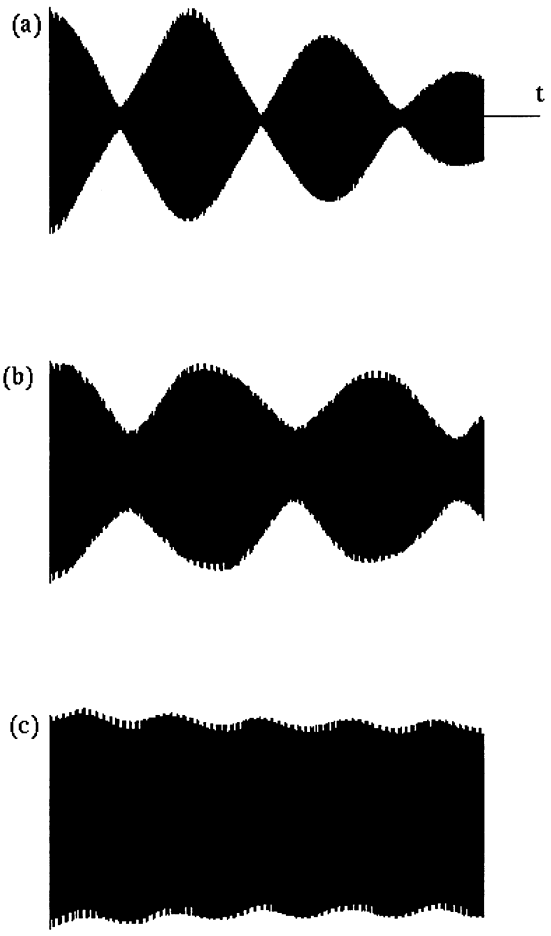


**Figure 4.** ICR time domain signals for the 6.5 ms detection time. **(a)** 50000, **(b)** 150,000, and **(c)** 350,000 ion simulations.

## *Image Charge Detection of Coherent Cyclotron Motion*

The ICR time domain signal is calculated from first principles during the course of the simulation [37]. Since the total potential is found at each time-step, the induced charge at a particular electrode point (a trap wall contains $64 \times 64$ electrode points) is proportional to the normal electric field at the electrode wall by Gauss's Law. Figure 4 plots ICR transients for the three different ion populations. These transients are the difference in total charge between two opposing trap walls that are parallel to the magnetic field as a function of time (i.e., differential detection). If coulomb interactions are negligible, then one expects to see three beat periods in the 6.5 ms transient corresponding to the number of times the two different masses pass each other due to their 459 Hz cyclotron frequency difference. A maximum in the transient occurs when the two ion clouds are in-phase (overlapping ion clouds) while a minimum occurs when they are at opposite sides of the trap. An inspection of Figure 4a shows the expected three beats for lowest ion population; however, slightly less than three beats are seen in Figure 4b (150k) and no clear beats are visible in Figure 4c (350k). Furthermore, the 50k transient decays in time while the higher population transients maintain their coherence to a much greater extent over the 6.5 ms detection time. This signal decay in Figure 4a arises from dephasing of the coherent cyclotron motion owing to differences in cyclotron frequency across the ion cloud [8, 34, 35]. Energy is well conserved in this simulation. The maximum variation in total energy for the detection period in Figure 4a is a fractional change of $5 \times 10^{-4}$. No ions are lost to the walls during detection for the 50k simulation. On the other hand, the higher ion population simulations (150k and 350k) had a considerable loss of ions to the radial walls. Furthermore, almost all of the ions lost to the walls were of just the higher cyclotron frequency species. The origin of this loss is discussed below.

Figure 5 displays Fourier transform magnitude spectra for the simulation time domain signals. At low density (Figure 5a) there are two cyclotron frequencies with nearly equal relative intensity. This is the expected coulomb interactions are negligible since the two different masses have equal relative abundance and are excited to the same coherent cyclotron radius. At higher density, Figure 5b, two mass peaks are still visible; however, the higher cyclotron frequency peak is considerably more intense than the lower frequency peak.

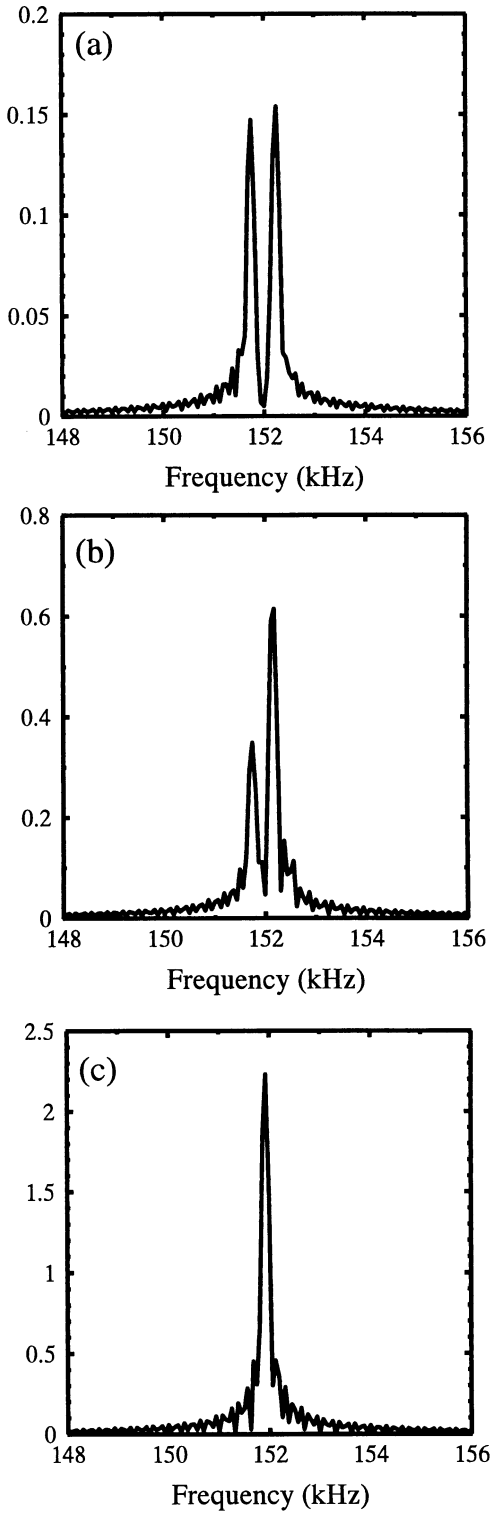


**Figure 5.** Fourier transform magnitude spectra of the transients from Figure 4.

At the highest number density, Figure 5c, there is only one frequency present indicating indirectly that cyclotron phase locking has occurred. The peak amplitudes are proportional to the number of ions in each mass peak. There are seven times more ions in the 350k simulation compared to the 50k simulation. Adding together the intensities of the two prominent peaks in Figure 5a yields about 0.3, which is about 1/7 the height of the single frequency peak in Figure 5c. This is the kind of information that a typical FTICR experiment routinely measures. The simulation has the advantage of visualizing directly the ion population.

Since the mass peaks are partially overlapping in Figure 5a, b, the actual peak positions may be shifted from their true location due to the non-additivity property of magnitude spectra [43, 44]. An alternative method is employed below to extract the cyclotron frequency from the simulation trajectories.

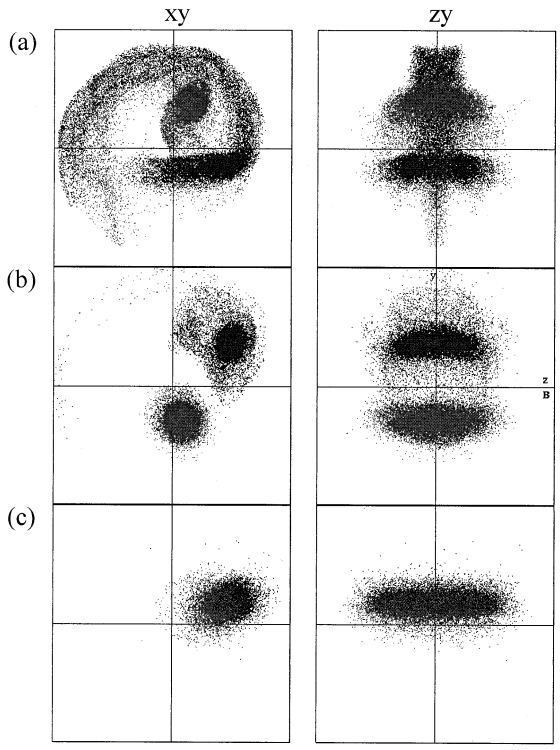


**Figure 6.** Snapshots in $xy$ perspective (left) and $zy$ perspective (right) at a late time during the detection period. **(a)** 50,000, **(b)** 150,000, and **(c)** 350,000 ions. The two masses have locked cyclotron modes in (c). The cubic trap boundary is indicated as the surrounding box.

## *Evolution of the Ion Ensemble*

Snapshots of the ion population are presented below at different stages of the simulation. Figure 6 displays particle positions in $xy$ (left side of Figure 6) and $zy$ (right side) perspective at a late time (~5 ms) in the detection event for the three different initial ion populations. The top, middle, and bottom frames in Figure 6 are snapshots from the 50k, 150k, and 350k simulations, respectively. Each simulation represents a qualitative change in the ion cloud evolution for these conditions. Starting with the 50k ion simulation (Figure 6a), the two

ion clouds, corresponding to the two different masses, lose cyclotron mode coherence over their trajectory. In contrast the two ion clouds for the 150k simulation (Figure 6b) maintain their coherence. Finally, at 350k only a single distribution is visible indicating that the two ion clouds have locked cyclotron modes. The particular time of the snapshots was selected to show two well-separated ion clouds if cyclotron phase locking does not occur. The two masses in Figure 6c for the 350k particle simulation are indistinguishable by FTICR image charge detection.

Inspection of the $yz$ perspectives (parallel to the magnetic field) show that at the highest number density, Figure 6c, the ion cloud maintains an ellipsoidal shape with a distinctive symmetry axis parallel to the magnetic field. In other words, neither the trap potential anharmonicity nor excitation potential nonlinearity have affected the shape of the ion cloud substantially. At medium number density, Figure 6b, the ion clouds are slightly distorted by the excitation nonlinearity but largely unaffected by trap anharmoncity. In Figure 6a, corresponding to the lowest number density, the trap anharmoncity results in a loss of cyclotron mode coherence over time while the excitation potential nonlinearity results in different $z$-amplitude ions receiving a different cyclotron radius resulting in a distortion of the pre-excitation ellipsoidal shape.

A more detailed visualization of the ion cloud evolution is depicted in Figures 7, 8, and 9 which are $xy$ perspectives of the ion distribution at different times during the 6.5 ms detection period. Snapshots are created on the fly by the PIC3D server as GIF images. At low number density (50k simulation) the individual ion clouds gradually lose cyclotron mode coherence as seen in Figure 7. At a sufficiently high number density (the 150k simulation in Figure 8) the two ion clouds maintain their coherence over the detection period; however, the lower cyclotron frequency (higher mass) ion cloud is more coherent than the higher frequency ion cloud. This disparity is due to the higher frequency cloud experiencing a greater time-averaged cyclotron radius than the lower frequency cloud as a result of the mutual coulomb interaction between the two clouds [7, 8]. Furthermore, as seen in Figure 8, the higher frequency ion cloud has a sufficiently large cyclotron radius at particular times (e.g., at 1.56 ms) to actually contact the radial trap wall resulting in ion and coherence loss. Finally, in Figure 9 corresponding to the highest density simulation, cyclotron phase locking is unambiguously observed since the two mass species do not separate completely into two ion clouds each with a different cyclotron frequency but rather remain locked together. It turns out that there are two partially overlapping distributions corresponding to the two different mass species in the single locked cloud in Figure 9. The higher mass maintains a near constant separation from the lower mass cloud, which is on the order of the individual cloud radius. Also, as seen in Figure 10 the higher mass species has a smaller coherent cyclotron radius than the lower mass cloud by this separation distance.

Figure 10 is a plot of the coherent cyclotron radius for each mass species (100.0 and 100.3 u) for the excitation and detection periods of all three simulations. Immediately after the excitation event, both masses have virtually the same coherent cyclotron radius and that this radius is the same for all three simulations. At the lowest number density (50k) the cyclotron radii modulate with a period near 2.2 ms and gradually decrease owing to loss of coherence. The medium density simulation (150k) maintains coherence with substantial modulation amplitude for the coherent cyclotron radii. As mentioned above, the higher frequency cloud (100.0 u) has a larger time-averaged cyclotron radius than the lower frequency ion cloud (100.3 u). Finally, at the highest number density simulation (350k) the two individual ion clouds have locked cyclotron modes resulting in a single detected cyclotron frequency. However, there is still a nonzero separation distance between cloud centers.

Figure 11 plots separation distance between the two centers of coherent cyclotron motion after cyclotron mode excitation. When the ion clouds are not locked (either 50,000 or 150,00 ions), the separation distance oscillates with a period close to the difference in cyclotron periods for the individual species, with an amplitude close to twice the cyclotron radius. On the other hand, when the clouds lock cyclotron modes (350,000 ions) the separation distance is about 0.1 cm, which is close to the cloud radius and much less than cyclotron radius of 0.6 cm. The realistic three-dimensional simulations show that the cyclotron separation distance for two phase locked ion clouds is nearly constant. This is in contrast to rigid ion cloud models that predict an oscillatory separation distance with relatively high frequency [2, 7, 8].

## *Comparison with a Stability Model*

The three simulations at different number density $n$ (number of ions per unit volume) depict qualitatively different evolutions. At lowest number density (with central density before excitation, $n/n_B = 0.04$) the two ion clouds dephase over the 6.5 ms detection period. At medium number density ($n/n_B = 0.18$) the clouds are coherent throughout the detection period and are not phase locked. At high number density ($n/n_B = 0.26$) the two ion clouds are coherent and phase locked. While hardly a complete survey of all possible dynamics, these results are explainable within the context of a relatively simple model.

A stability theory, developed by Peurrung and Kouzes [6], for a single mass to charge ratio ion cloud with a coherent cyclotron mode predicts that the ion cloud is more stable, meaning a longer lasting ICR transient, the higher the number density. This model explains qualitatively why an ion cloud in a relatively anharmonic trap such as the cubic trap can have detected signals lasting more than $10^6$ cyclotron periods.

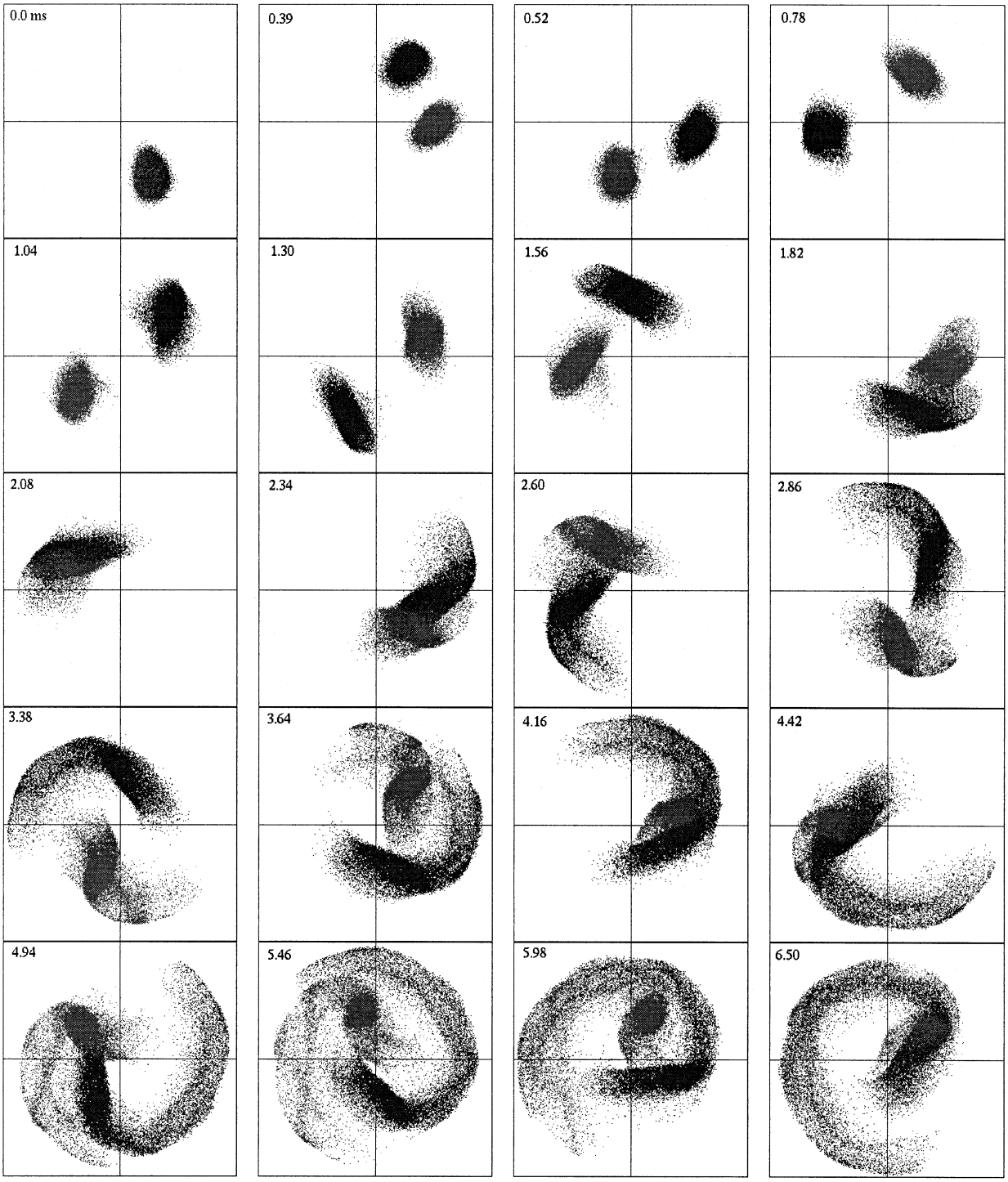


**Figure 7.** Snapshots of ion *xy* positions for the 50,000 ion simulation at selected times during the detection period. The cubic trap boundary is indicated.

Stability conditions were deduced using dimensional and numerical analysis for a single test charge interacting with a rigid ion cloud that has a different cyclotron frequency than the test charge [6]. This cyclotron frequency difference can be due to any physical process such as trap potential anharmonicity or magnetic field inhomogeneity, or due to a mass difference between test ion and ion cloud. Basic approximations of this stability model include that the ion cloud has a constant density, circular cross section and a cyclotron radius much greater than the cloud radius. The same parameter dependencies are found in computer simulations

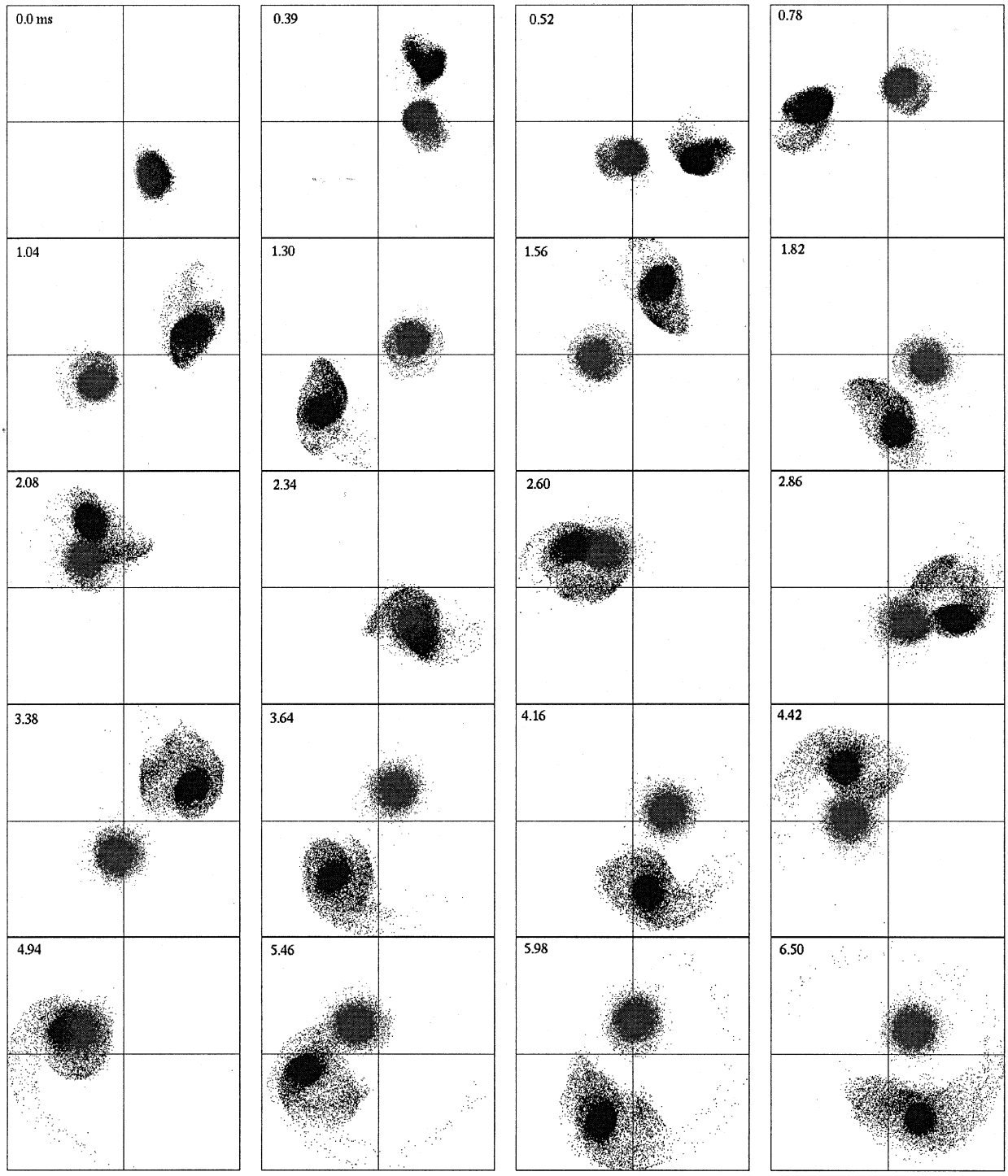


**Figure 8.** Snapshots of ion $xy$ positions for the 150,000 ion simulation at selected times during the detection period.

on the dynamics of two similar mass rigid ion clouds [7, 8]. Peurrung and Kouzes originally derived the stability condition as a minimum required plasma frequency divided by cyclotron frequency ratio [6]. Rewriting their stability condition in terms of a minimum required number density to Brillouin density yields

$$\frac{n}{n_B} > \frac{18}{a(\alpha)}\left(\frac{R_c}{\rho_c}\right)\left(\frac{\Delta\omega_c}{\omega_c}\right) \approx 13\left(\frac{R_c}{\rho_c}\right)\left(\frac{\Delta\omega_c}{\omega_c}\right) \quad (1)$$

where $n$ is the number density and $n_B = \varepsilon_0 B^2/2m$ is the maximum achievable trap number density. $R_c$, $\rho_c$,

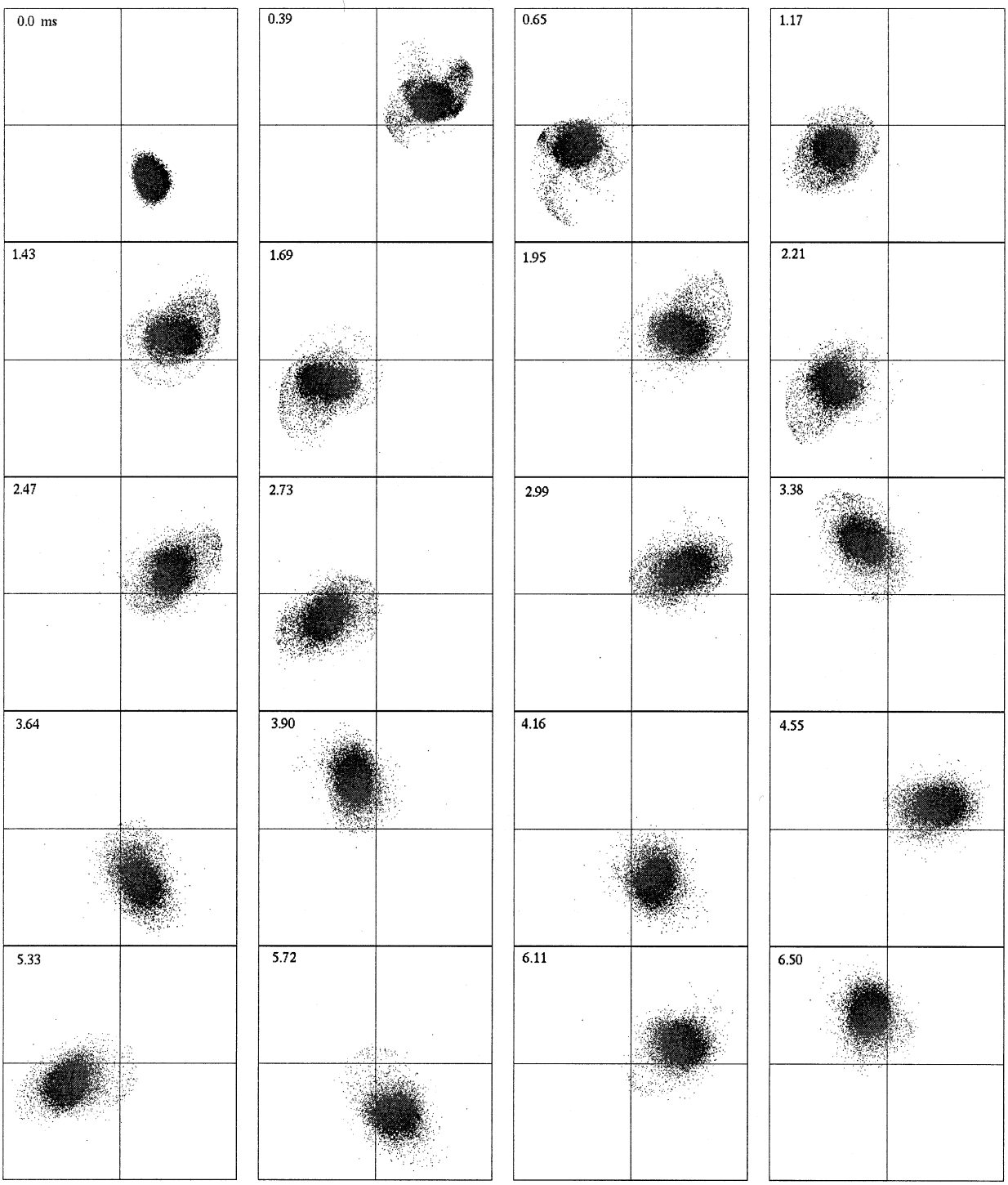


**Figure 9.** Snapshots of ion $xy$ positions for the 350,000 ion simulation at selected times during the detection period.

$\omega_c$, and $\Delta\omega_c$ are the cyclotron radius, cloud radius, cyclotron frequency, and difference in cyclotron frequency, respectively. The $a(\alpha)$ is a dimensionless number dependent of the cloud aspect ratio [6]. Constant density spherical and infinitely long cylindrical clouds have $a(\alpha)$ equal to 1 and 1.5, respectively [6]. The factor of 13 in eq 1 is derived from earlier simulation work on two interacting rigid cylindrical ion clouds [8]. The stability condition is that $n/n_B$ must be greater than the right hand side of eq 1 in order for the ion cloud to remain coherent or locked. One should note that the minimum number density $n$ to achieve stability is

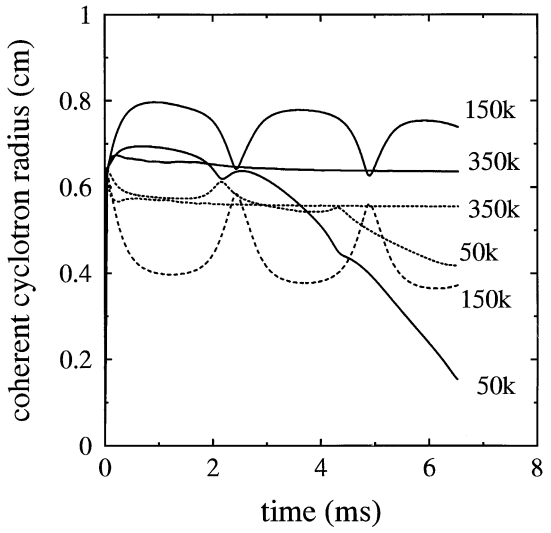


**Figure 10.** Evolution of the coherent cyclotron radii during the detection period for the two masses. **(a)** 50,000, **(b)** 150,000, and **(c)** 350,000 ion simulations. The 100.0 and 100.3 u mass species are plotted as solid and broken lines, respectively.

proportional to $B^2$ since $n_B$ is proportional to $B^2$. The advantage of defining the stability condition eq 1 in terms of $n/n_B$ is that $n/n_B$ is limited from a minimum of zero to a maximum of exactly one.

In the three simulations the ratio of cyclotron radius to cloud radius is $R_c/\rho_c \cong 5$. For cyclotron phase locking, the difference in cyclotron frequency is due to the mass difference between the two ion clouds, which gives $\Delta\omega_c/\omega_c = \Delta m/m = 0.003$. Using these parameters in eq 1 yields a minimum number density required for cyclotron phase locking of $n/n_B \cong 0.20 \pm 0.05$. This compares well with the simulation results using the number density before cyclotron mode excitation.

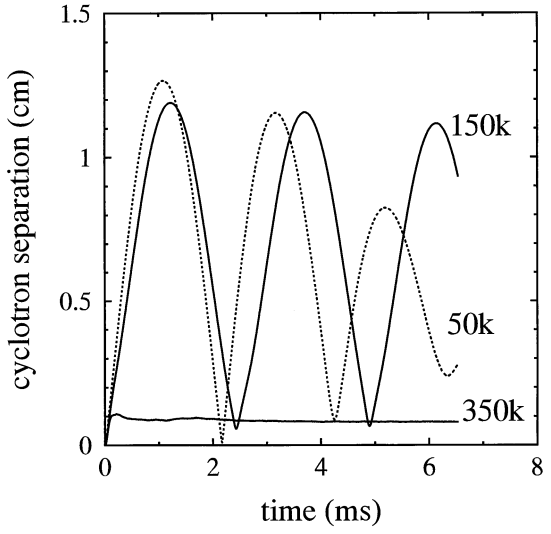


**Figure 11.** The separation distance between the centers of coherent cyclotron motion for the two masses for the 50,000, 150,000, and 350,000 ion simulations.

The ion clouds are locked in the 350k simulation but not locked in the 150k simulation where the number density ratios $n/n_B$ immediately before excitation are 0.26 and 0.18, respectively. These are the central densities.

Additional simulations were carried out at several other number densities in order to constrain the phase locking threshold. Phase locking occurs when 200,000 ions were used, which yielded $n/n_B = 0.20$ for the central density just before cyclotron excitation.

Equation 1 also provides a partial explanation of the different evolutions observed for the low (Figure 7) and medium (Figure 8) number density simulations. Assuming that the predominant mechanism to dephase an ion cloud at low ion populations is from the cubic trap potential anharmonicity, then $\Delta\omega_c$ can be estimated from the frequency shift due to the fourth-order trap potential [8]

$$\delta\omega_c = \frac{3V_tD_4}{2Bd^4}(-2A_z^2 + 2R_m^2 + R_c^2) \quad (2)$$

where $A_z$ and $R_m$ are the $z$ amplitude and magnetron radius of a single ion, respectively. For a cubic trap, $D_4 = 1.02$. Following the same procedure described in earlier work [8], $\Delta\omega_c$ from the trap anharmonicity is approximately the difference in cyclotron frequency between an ion located at the cloud center ($A_z = 0$) and an ion which traverses the entire cloud length ($A_z \cong 0.5$ cm). Using these values along with the simulation parameters listed in Table 1 in eq 2 yields a frequency spread of $\Delta\omega_c \cong (2\pi)$ 30 Hz across the ion cloud. In the absence of coulomb interactions the ion cloud dephases in ~30 ms. Figure 7 shows that the ion clouds dephase in a time ~10 ms indicating that coulombic effects may be important as well as trap anharmonicity in this lower density simulation. Along with the other simulation parameters in eq 1, the minimum number density required to achieve stability against trap potential anharmonicity is $n/n_B \cong 0.013 \pm 0.003$. This is the predicted minimum number density required to stabilize cyclotron mode coherence in each ion cloud. Before cyclotron mode excitation, the ion population consists of a single cloud containing both mass species aligned along the trap axis. After cyclotron excitation, the ion clouds separate into different lower density ion clouds with densities dependent upon the relative abundance of each species. For all simulations, two mass species are present in equal relative abundance. Before cyclotron excitation, the central (maximum) number densities for the 50k and 150k simulations were measured as $n/n_B = 0.04$ and 0.18, respectively. Therefore, after cyclotron excitation the single ion cloud separates into two individual ion clouds each with densities of $n/n_B = 0.02$ and 0.09 for the 50k and 150k particle simulations, respectively. Since the 50k simulation (Fig. 7) result of $n/n_B = 0.02$ for each cloud is greater than the predicted stability requirement of $n/n_B \cong 0.013 \pm$

0.003, the stability model described by eq 1 with eq 2 is only qualitatively correct in this case.

### *Frequency Shifts*

A FTICR experiment determines mass to charge ratio by measuring cyclotron frequency. This section reports simulation results for the frequency of coherent cyclotron motion at 1 T for the 100.0 and 100.3 u mass species. Five different simulation runs were carried out including the 50k, 150k, and 350k simulations described above, as well as simulations with 250,000 and 500,000 ions. The 500,000 ion simulation actually used 250,000 simulation particles with each simulation particle representing two single ions (super = 2, see Table 1). A precise determination of the cyclotron frequency shift is obtained from the simulation results, not by Fourier transforming the detected image charge, but rather by counting the number of times the position of coherent cyclotron motion revolves around the trap center during the detection time. This approach also eliminates the uncertainty in peak position for closely spaced magnitude peaks due to nonadditivity of magnitude spectra [43, 44].

The algorithm to count the number of cyclotron cycles that each mass species executes during the 6.5 ms detection period relies first on determining the cyclotron position ($x_c$, $y_c$) from the ion perpendicular velocity components ($v_x$ and $v_y$) calculated during the simulation. The radial motion position vector is the sum of the cyclotron and guiding center (magnetron) position vectors. For the simulation conditions, after the excitation event resulting in a substantial coherent cyclotron radius, the magnitude of the cyclotron velocity is much larger than the drift (magnetron) velocity by a factor on the order of one thousand. Furthermore, taking an average over all ions of a particular mass automatically removes the velocity contribution due to coulomb induced internal cloud rotation [6, 37], leaving just the center of mass motion. Therefore, the ion perpendicular velocity components, $v_x$ and $v_y$, are approximately equal to the cyclotron velocity components, and

$$x_c \cong \frac{-v_y}{\omega_c} \quad \text{and} \quad y_c \cong \frac{+v_x}{\omega_c} \tag{3}$$

Taking an average of eq 3 over all simulation particles of a particular mass gives the coherent cyclotron position $X_c$, $Y_c$. The small difference between $\omega_c$ and the actual shifted (perturbed) cyclotron frequency results in an error of about 1% in the location of the true coherent cyclotron position. Once $X_c$ and $Y_c$ are found, the coherent cyclotron radius is $R_c = (X_c^2 + Y_c^2)^{0.5}$ and the coherent cyclotron phase $\phi_c = \tan^{-1}(Y_c/X_c)$. The error in calculating $R_c$ is the same as for determining $X_c$ and $Y_c$, namely equal to the error between $\omega_c$ and the actual shifted cyclotron frequency. However, this error does not affect the phase since the ratio $Y_c/X_c$ depends only on the ratio of ion perpendicular velocity components. If the coherent cyclotron frequency does not change appreciably during the detection period, a plot of cyclotron phase $\phi_c$ versus time results in a straight line fit whose slope equals the shifted cyclotron frequency. Furthermore, the standard deviation ($\sigma$) in the fitted slope from linear regression gives an error estimate for the uncertainty in the measured coherent cyclotron frequency.

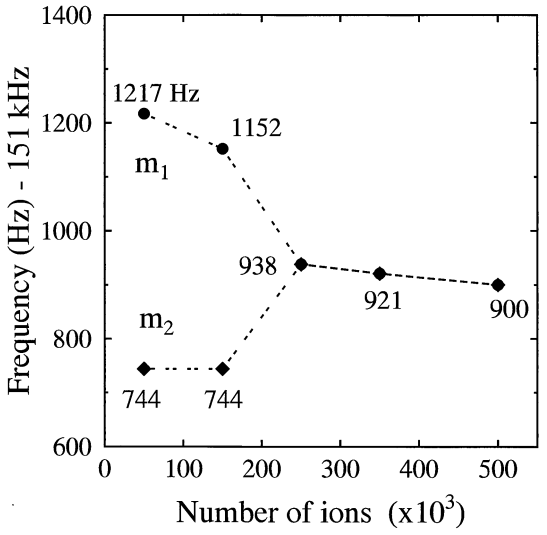


**Figure 12.** The frequency of coherent cyclotron motion, for the two species $m_1$ (100.0 u) and $m_2$ (100.3 u), as a function of the total number of confined ions. Results for five different simulations are shown with number of ions ranging from 50,000 to 500,000. Cyclotron phase locking occurs when the two masses have the same detected cyclotron frequency. The uncertainty is ±2 Hz. The lines are only meant to guide the eye. Other simulation parameters are listed in Table 1.

Figure 12 is a plot of the perturbed cyclotron frequencies for five simulation runs with total number of ions ranging from 50,000 to 500,000. The cyclotron frequencies are calculated from the slope of a straight-line least squares fit of the coherent cyclotron phase versus time. The $2\sigma$ uncertainty in these fits is less than ±2 Hz, which is taken as the uncertainty in the calculated cyclotron frequencies. While the absolute frequencies obtained from the particle-in-cell simulations using the leap-frog integrator with finite time-step does not give the absolute cyclotron frequency to an accuracy of ±2 Hz, differences in cyclotron frequency are much more precise.

In order to demonstrate that this algorithm gives correct results for calculated frequencies, a comparison was made with the usual FT method on the detected time domain signal. The time domain signal shown in Figure 4c for the 350,000 ion simulation was used for this purpose. This time domain signal was zero filled a sufficient number of times to give 8 Hz difference between frequency domain points. A quadratic fit to the magnitude spectrum peak centroid gave a peak location of 151,922.4 Hz. This is within uncertainty to the result

of 151,921 ± 2 Hz, displayed in Figure 12 for the 350,000 ion simulation, confirming the validity of the cyclotron phase algorithm for calculating the frequency of coherent cyclotron motion.

Starting with the 50,000 ion simulation, from Figure 12 the difference in shifted cyclotron frequency between $m_1$ (100.0 u) and $m_2$ (100.3 u) is 473 ± 4 Hz, which compares reasonably well with the difference in unperturbed cyclotron frequencies $\Delta\omega_c = (2\pi)$ 459 Hz. Increasing the number of ions to 150,000, $m_1$ has a lower cyclotron frequency compared to the 50k simulation by 65 Hz, while there is no change in the cyclotron frequency for $m_2$. This constancy of cyclotron frequency for $m_2$ with increasing number of ions, before phase locking occurs, is explainable by a simple model. Figure 10 shows that for the 150,000 ion simulation the coherent cyclotron radius for the higher cyclotron frequency species ($m_1$) is greater than the lower frequency species ($m_2$). Earlier work [7], using a line charge model for the charge distribution, has shown that when the cyclotron radii are different, the smaller radius species receives zero average coulomb force from the larger radius species. The large downward shift for $m_1$ of 65 Hz in going from 25,000 ions in each mass (i.e., 50,000 total ions) to 75,000 ions comes from at least two sources; the coulomb interaction from $m_2$, and the image charge interaction.

Both mass species have the same shifted cyclotron frequency, hence are phase locked, when the total number of ions is at least 200,000. The three largest ion population simulations in Figure 12 correspond to 250,000, 350,000, and 500,000 ions. In going from 250,000 to 500,000 ions there is a linear downward shift of 38 ± 4 Hz. This shift is attributable to the image charge interaction acting on the phase locked ion cloud. Gorshkov et al. [22] derived the image charge frequency shift $\delta\omega_c$ for a line charge with linear charge density $Nq/L$ inside a cylindrical trap (length $L$, radius $r_{\text{trap}}$)

$$\delta\omega_c = \frac{-Nq}{2\pi\varepsilon_0 LB(r_{\text{trap}}^2 - R_c^2)}. \quad (4)$$

Using eq 4 with $N$ = 250,000, $q = e$, $L$ = 0.025 m, $B$ = 1 T, $r_{\text{trap}}$ = 0.0125 m, and $R_c$ = 0.006 m, corresponding to simulation conditions after phase locking has been reached, gives an image charge frequency shift of $\delta\omega_c = -(2\pi)$ 38 Hz. The exact agreement between eq 4 and the simulation result is somewhat fortuitous since the trap used in the simulations is cubic while eq 4 is based on an infinitely long cylinder. Also, eq 4 is based on a line charge approximation for the distribution while the simulation distributions were ellipsoidal and did not extend the entire trap length. Nonetheless, the model calculation demonstrates the most of the frequency shift after cyclotron phase locking is due to the image charge interaction.

Finally, the frequency at phase locking for the 250,000 ion simulation of 938 Hz (+151 kHz) is not a simple average of the two unperturbed cyclotron frequencies. The lowest number density simulation (50,000 ions) has an average frequency of 0.5(1217 + 744) = 980 Hz, which is 42 ± 4 Hz higher than the frequency at phase locking when there are 250,000 ions. Most of this difference between average frequency at very low density and the frequency just after phase locking occurs is accountable from the expected downward shift of ≈30 Hz due to image charge interaction.

### *Magnetic Field Dependence*

The magnetic field, cyclotron radius and mass dependencies of the minimum number density required to cause two close mass peaks to lock cyclotron modes are of considerable importance to analytical FTICR-MS, especially at high molecular mass [8]. Earlier simulation work [8] predicts a $B^2$ (magnetic field squared) dependence for the minimum number density required to lock cyclotron modes between two close masses in agreement with eq 1. PIC3D is applied at three different magnetic fields (0.5, 1.0, and 2.0 T) in order to test the $n \propto B^2$ prediction under realistic conditions. At 1 T, the same simulation parameters listed in Table 1 are employed except the number of simulation particles are varied in different simulation runs in order to constrain the minimum $n/n_B$ required to cause cyclotron phase locking. It is found that when $n/n_B = 0.20$ the two masses (100.0 and 100.3 u) locked modes; however, when $n/n_B = 0.18$ phase locking did not occur. The density $n$ is measured as the maximum density in the distribution (the central density) just before cyclotron mode excitation.

At a magnetic field of 2 T, the trap potential was increased from 1 to 2 V in order to confine a sufficient number of ions to cause cyclotron phase locking. At 1 V the trap (width 2.5 cm) filled to capacity with ions before cyclotron phase locking between the 100.0 and 100.3 u mass ions occurred. In addition, by increasing the trap potential the cloud aspect ratio was similar to the 1 T simulation run. The time-step, excitation frequency, and excitation voltage were appropriately modified to attain equivalent excitation and detection conditions as for the 1 T simulations. With these parameters, phase locking occurred when $n/n_B = 0.18$ but not when $n/n_B = 0.17$. Finally, at a magnetic field of 0.5 T, the trap potential was reduced to 0.5 V in order to have a comparable cloud aspect ratio as for the other two magnetic field values. At 0.5 T, the masses locked cyclotron modes when $n/n_B = 0.22$ but not when $n/n_B = 0.20$.

Since $n/n_B$ is practically constant at all three magnetic fields and the density limit $n_B$ is directly proportional to $B^2$, the minimum number density $n$ required to cause phase locking is proportional to $B^2$ in agreement with earlier work. Figure 13 is a plot of the PIC3D results for the three different magnetic fields. These results are compared in Figure 13 to three different assumed magnetic field dependencies demonstrating that $n \propto B^2$

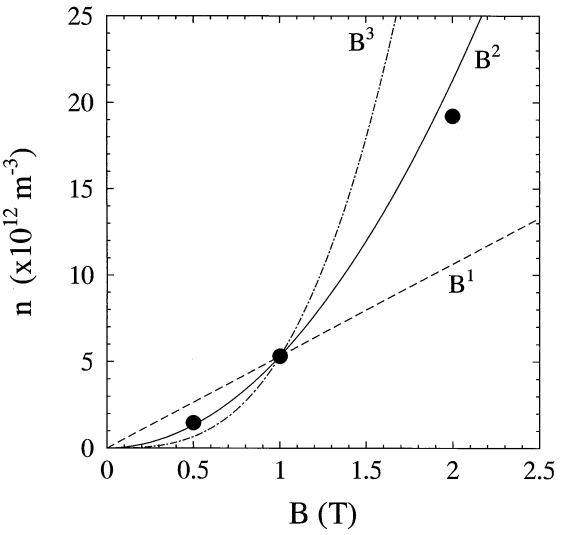


**Figure 13.** Magnetic field dependence of the minimum number density required to lock cyclotron modes. The solid circles are the simulation results at three different magnetic fields. The lines are three different assumed magnetic field dependencies, $n$ proportional to $B^1$, $B^2$, and $B^3$.

is a much better fit than either $n \propto B^1$ or $n \propto B^3$, for the minimum number density required to lock cyclotron modes.

## Conclusions

PIC3D is a versatile, fully three-dimensional many particle simulation code for modeling complete experimental sequences in ion cyclotron mass spectrometers. The PIC3D Java client is the graphical user interface, which sets up, controls and views in a real-time, a simulation running on a different computer that runs the PIC3D server. The PIC3D server is the computational engine that executes the three-dimensional particle-in-cell algorithm and communicates with requesting clients such as the PIC3D Java client. For long running simulations, the remote user disconnects the client-server socket communication; however, the real-time status of a simulation in progress is viewable at anytime by using an ordinary web browser.

As a first application, PIC3D is applied to the problem of coherent cyclotron motion of two close masses at high ion population. Each simulation run involves solving Poisson's equation on a $64 \times 64 \times 64$ grid at each time-step for more than 100,000 time-steps. Poisson's equation is solved with the exact cubic trap boundary conditions with user specified static and/or time-dependent potentials. The ion population range from 50,000 to 350,000 coulombically interacting particles confined in a cubic ICR trap. A four step simulation sequence is carried out consisting of sequential steps of ion loading into the trap, collisional cooling, cyclotron mode excitation and image charge detection. These are the most realistic simulations to date of large numbers of interacting ions in an ICR trap. At low ion density the ion clouds lose coherence owing to differences in cyclotron frequency across the cloud. At medium density the clouds remain coherent and are not locked. At high number density the two clouds lock cyclotron modes. Simulations at different magnetic fields show a $B^2$ dependence for the minimum density required to lock cyclotron modes. Future work will address the mass and cyclotron radius dependencies of cyclotron phase locking.

## Acknowledgment

I thank Dr. R. B. Rex, at EMSL, for early discussions on possible ways to create an Internet accessible user interface for a many particle simulation code.

## References

1. Huang, J.; Tiedemann, P. W.; Land, D. P.; McIver, R. T.; Hemminger, J. C. *Int. J. Mass Spectrom. Ion Processes* **1994,** *134*, 11–21.
2. Naito, Y.; Inoue, M. *J. Mass Spectrom. Soc. Jpn.* **1994,** *42*, 1–9.
3. Pasa-Tolic, L.; Huang, Y.; Guan, S.; Kim, H. S.; Marshall, A. G. *J. Mass Spectrom.* **1995,** *30*, 825–833.
4. Anderson, J. S.; Laude, D. A. *Int. J. Mass Spectrom. Ion Processes* **1996,** *158*, 163–174.
5. Stults, J. T. *Anal. Chem.* **1997,** *69*, 1815–1819.
6. Peurrung, A. J.; Kouzes, R. T. *Phys. Rev. E* **1994,** *49*, 4362–4368.
7. Mitchell, D. W.; Smith, R. D. *Phys. Rev. E* **1995,** *52*, 4366–4386.
8. Mitchell, D. W.; Smith, R. D. *J. Mass Spectrom.* **1996,** *31*, 771–790.
9. Comisarow, M. B.; Marshall, A. G. *Chem. Phys. Lett.* **1974,** *25*, 282–283.
10. Comisarow, M. B. *J. Chem. Phys.* **1978,** *69*, 4097–4104.
11. Marshall, A. G.; Verdun, F. R. *Fourier Transforms in NMR, Optical and Mass Spectrometry: A User's Handbook*; Elsevier: Amsterdam, 1990.
12. Marshall, A. G.; Grosshans, P. B. *Anal. Chem.* **1991,** *63*, 215A.
13. Asamoto, B.; Dunbar, R. C. *Analytical Applications of Fourier Transform Ion Cyclotron Resonance Mass Spectrometry*; VCH: New York, 1991.
14. Comisarow, M. B. *Adv. Mass Spectrom.* **1980,** *8*, 1698–1706.
15. Comisarow, M. B. *Int. J. Mass Spectrom. Ion Processes* **1981,** *37*, 251–257.
16. Grosshans, P. B.; Marshall, A. G. *Int. J. Mass Spectrom. Ion Processes* **1990,** *100*, 347–379.
17. Guan, S.; Marshall, A. G. *Int. J. Mass Spectrom. Ion Processes* **1995,** *146/147*, 261–296.
18. Dienes, T.; Pastor, J.; Schurch, S.; Scott, J. R.; Cui, S.; Wilkens, C. L. *Mass Spectrom. Rev.* **1996,** *15*, 161–211.
19. Jeffries, J. B.; Barlow, S. E.; Dunn, G. H. *Int. J. Mass Spectrom. Ion Processes* **1983,** *54*, 169–187.
20. Chen, S.-P.; Comisarow, M. B. *Rapid Commun. Mass Spectrom.* **1991,** *5*, 450–455.
21. Chen, S.-P.; Comisarow, M. B. *Rapid Commun. Mass Spectrom.* **1992,** *6*, 1–3.
22. Gorshkov, M. V.; Marshall, A. G.; Nikolaev, E. N. *J. Am. Soc. Mass Spectrom.* **1993,** *4*, 855–868.
23. Herold, L. K.; Kouzes, R. T. *Int. J. Mass Spectrom. Ion Processes* **1990,** *96*, 275–289.
24. Hearn, B. A.; Watson, C. H.; Baykut, G.; Eyler, J. R. *Int. J. Mass Spectrom. Ion Processes* **1990,** *95*, 299–316.

25. Xiang, X.; Guan, S.; Marshall, A. G. *J. Am. Soc. Mass Spectrom.* **1994,** *5*, 238–249.
26. Guan, S.; Pasa-Tolic, L.; Marshall, A. G.; Xiang, X. *Int. J. Mass Spectrom. Ion Processes* **1994,** *139*, 75–86.
27. Naito, Y.; Inoue, M. *Rapid Commun. Mass Spectrom.* **1997,** *11*, 578–586.
28. Londry, F. A.; Alfred, R. L.; March, R. E. *J. Am. Soc. Mass Spectrom.* **1993,** *4*, 687–705.
29. Julian, R. K.; Nappi, M.; Weil, C.; Cooks, R. G. *J. Am. Soc. Mass Spectrom.* **1995,** *6*, 57–70.
30. Dahl, D. A. *Simion 3D Version 6.0 User's Manual*, 1995.
31. Wang, T.-C. L.; Marshall, A. G. *Int. J. Mass Spectrom. Ion Processes* **1986,** *68*, 287–301.
32. Uechi, G. T.; Dunbar, R. C. *J. Am. Soc. Mass Spectrom.* **1992,** *3*, 734–741.
33. Peurrung, A. J.; Kouzes, R. T. *Int. J. Mass Spectrom. Ion Processes* **1995,** *145*, 139–153.
34. Miluchihin, N. V.; Miura, K.; Inoue, M. *Rapid Commun. Mass Spectrom.* **1993,** *7*, 966–970.
35. Nikolaev, E. N.; Miluchihin, N. V.; Inoue, M. *Int. J. Mass Spectrom. Ion Processes* **1995,** *148*, 145–157.
36. Han, S.-J.; Shin, S. K. *J. Am. Soc. Mass Spectrom.* **1996,** *8*, 319–326.
37. Mitchell, D. W.; Smith, R. D. *Int. J. Mass Spectrom. Ion Processes* **1997,** *165/166*, 271–297.
38. Birdsall, C. K.; Langdon, A. B. *Plasma Physics via Computer Simulation*; McGraw-Hill: New York, 1985.
39. Hockney, R. W.; Eastwood, J. W. *Computer Simulation Using Particles*; Adam Hilger: New York, 1988.
40. Birdsall, C. K. *IEEE Trans. Plasma Sci.* **1991,** *19*, 65–85.
41. Brillouin, L. *Phys. Rev.* **1945,** *67*, 260–266.
42. Guo-Zhong, Li; Guan, S.; Marshall, A. G. *J. Am. Soc. Mass Spectrom.* **1998,** *9*, 473–481.
43. Lee, J. P.; Chow, K. H.; Comisarow, M. B. *Anal. Chem.* **1988,** *60*, 2212–2218.
44. Chow, K. H.; Comisarow, M. B. *Int. J. Mass Spectrom. Ion Processes* **1989,** *89*, 187–203.