LA-12628-MS

UC-000
Issued: December 1993

Particle-In-Cell (PIC) Code Simulation Results and Comparison with Theory Scaling Laws for Photoelectron-Generated Radiation

Thomas M. Dipp[*]

[*] *Military Research Associate, Los Alamos National Laboratory, USAF AFOSR/NI, Building 410, Bolling AFB, D.C. 20332-6448.*

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

Contents

List of Figures

List of Tables

PARTICLE-IN-CELL (PIC) CODE SIMULATION RESULTS AND COMPARISON WITH THEORY SCALING LAWS FOR PHOTOELECTRON-GENERATED RADIATION

by

Thomas M. Dipp*

ABSTRACT

The generation of radiation via photoelectrons induced off of a conducting surface was explored using Particle-In-Cell (PIC) code computer simulations. Using the MAGIC PIC code, the simulations were performed in one dimension to handle the diverse scale lengths of the particles and fields in the problem. The simulations involved monoenergetic, nonrelativistic photoelectrons emitted normal to the illuminated conducting surface. A sinusoidal, 100% modulated, 6.3263 ns pulse train, as well as unmodulated emission, were used to explore the behavior of the particles, fields, and generated radiation. A special postprocessor was written to convert the PIC code simulated electron sheath into far-field radiation parameters by means of rigorous retarded time calculations. The results of the small-spot PIC simulations were used to generate various graphs showing resonance and nonresonance radiation quantities such as radiated lobe patterns, frequency, and power. A database of PIC simulation results was created and, using a nonlinear curve-fitting program, compared with theoretical scaling laws. Overall, the small-spot behavior predicted by the theoretical scaling laws was generally observed in the PIC simulation data, providing confidence in both the theoretical scaling laws and the PIC simulations.

*Military Research Associate, Los Alamos National Laboratory,
USAF AFOSR/NI, Building 410, Bolling AFB, D.C., 20332-6448

Introduction

This paper summarizes the computational results obtained from about 50 Particle-In-Cell (PIC) simulations. Most of these simulations were performed during a two-month period, and occupied several Sun Sparc workstations for a total of approximately 100 SparcStation2 days. These simulations provide a glimpse into the electron sheath dynamics and radiation generated from a monoenergetic, perpendicular electron beam emitted off of a conducting plate. First I will discuss the simulations and present typical graphs of the data obtained. Next, I will analytically compare the PIC data to my theory equations, which were presented in a previous paper [Ref. 1]. Lastly, I will briefly discuss the major conclusions. Finally, unless stated otherwise, almost all equations, parameters, and constants used in the paper are expressed in the International Metric System (SI) system of measurement.

All of the PIC code simulation results *presented* in this paper were performed using Mission Research Corporation's (MRC) MAGIC PIC code [Ref. 2]. However, some of the PIC simulations were repeated using the Los Alamos National Laboratory (LANL) MERLIN PIC code [Ref. 3] and LANL Cray supercomputers to confirm the MAGIC PIC code simulation results. Further, most of the MAGIC PIC code simulations were performed using Sun workstations at the Phillips Laboratory, though some MAGIC PIC code simulations were also performed using Cray supercomputers at LANL. The results presented and analyzed in the main body of the paper, as well as the data listed in **Appendix A**, were all obtained from MAGIC PIC code simulations performed on Sun workstations. However, **Appendix B** summarizes some of the additional MAGIC PIC code simulation results generated on Cray supercomputers at LANL.

The MAGIC PIC code is a commercial, two-dimensional, finite-difference, time-domain, plasma simulation code. Based on the initial state of a problem involving various boundary conditions, fields, particles, and such, the code simulates the problem's physics and evolves the solution in time. MAGIC uses the complete Lorentz force equation to push the particles and Maxwell's time-dependent equations to obtain the fields. MAGIC performs this iterative particle-pushing and field-solving process self-consistently by communicating the particle and field component information through the simulation grid. As such, MAGIC and other PIC codes obtain numerical solutions through a basic approach that uses fundamental physics equations but few approximations or simplifying models to operate directly on discrete particles. However, pushing lots of particles through many cells of a simulation grid (even when using large 'macro particles') self-consistently with each other and the boundary conditions does have its drawbacks, requiring large amounts of computer memory and central processing unit (CPU) time. Still, MAGIC was used for this problem because PIC codes are robust, workstation CPU time is inexpensive, and the MAGIC PIC code was a well-known, proven code.

PIC Code Simulations

The PIC code simulations and results will be presented in this section. First the PIC code simulations and the parameters that were explored will be discussed. Finally, various PIC code simulation results will be displayed and discussed. Most of the results in this section will be presented as graphs, though the numerical data used in some of the graphs are listed in **Appendix A**. Also, equations will be presented to facilitate converting the radiated energy, as calculated by the PIC code using a beam emission model and PIC code parameters, into photon or laboratory parameter set equivalents. In summary, this section will present a broad range of information covering the details of the PIC code simulations and the behavior of various observables.

Background

All of the MAGIC simulations used in this paper share several common features:

1. One-dimensional simulations which reproduce 'small-spot' effects.
2. Beam emission area of 2.0×10^{-6} square meters in the $x - y$ plane.
3. Emitted electron energy that is constant, uniform, and monoenergetic.
4. Beam current density that is either constant or 100% modulated as a pure sine-squared type of pulse-train at a single frequency.
5. Cartesian coordinate system for the simulation grid.
6. A rectangular simulation grid whose boundaries, from left to right, consist of a vertical conducting plate, two horizontal mirror symmetry boundary conditions, and a vertical outgoing wave boundary condition. The grid is 2 cells high by 100 cells long in the $x - z$ plane. As MAGIC is a $2\frac{1}{2}$ dimensional PIC code, symmetry also exists about the $x - z$ simulation's sides, along the y-axis.
7. Two micron (x-axis) by one meter (y-axis) conducting beam emission surface, spanning two cells along the grid's left x-axis boundary. The

one meter length along the y-axis is the default extent assigned to the two-dimensional $x - z$ simulation plane.

8. Perpendicular beam emission off the conducting plate, along the z-axis.

9. One hundred microns of simulated vacuum located perpendicular to the conducting surface, consisting of 100 cells in length along the z-axis.

10. Problems were simulated for 6.3263 nanoseconds.

One-dimensional simulations were necessary to realistically handle the problem's diverse scale lengths involving particle dynamics, field propagation speeds, and radiation wavelengths. Realistic problem simulations need to be self-consistent, reproduce actual small or large spot effects, and run for realistic simulation times to reach steady-state behavior and provide reasonable frequency resolution when Fast Fourier Transforms (FFTs) are employed. For example, a 2-D PIC simulation made possible by using cells with huge aspect ratios loses self-consistency and is similar to running many 1-D PIC simulations simultaneously and synthesizing the results. Yet 2-D PIC simulations that are self-consistent will still basically model 1-D small-spot diffraction effects, and generally can only be simulated for such short times that even reaching steady state is hard, while achieving good FFT resolution becomes even harder.

The above becomes clear when one realizes that 2-D PIC simulations that are self-consistent are limited in aspect ratio, so that the longitudinal cell size needed to resolve the particle dynamics, being quite small, limits the radial cell size as well. Moreover, the small cell dimensions would then require a correspondingly small time step due to Courant constraints imposed by the PIC code in order to simulate the physical system being modeled. Therefore, in order to then run the simulation for a realistic 6.3263 ns, where steady state and a reasonable frequency resolution are obtained, around one million plus time steps would be required. Consequently, this leads to a 100 x 10000 cell grid using a 4-to-1 aspect ratio and running for a million time steps to perform a realistic typical 2-D run. Such a 2-D PIC simulation would not only require large amounts of memory, but impossible amounts of CPU time by several orders of magnitude (see **Appendix C**).

Clearly, 1-D runs which can provide much of the physics for small spots, and some of the physics for large spots, are desirable. Even with 1-D simulations, various common settings were used in MAGIC to help speed up the PIC run times. These common algorithm choices were: particle-to-field step subcycling, electric field only calculations (no magnetic field components need to be calculated in a 1-D problem), and nonrelativistic particle dynamics. Combining 1-D problems with the above speed up techniques resulted in an average run time per MAGIC simulation of 1.5-to-2 days on a 32-bit Sparc2 workstation rated at two million calculations per second. Lastly, good throughput was achieved by writing UNIX script files to automatically and continuously run and postprocess many parameter simulations to generate data for various graphs on many different workstations.

MAGIC simulations explored several theory parameters from my theory paper [Ref. 1]. Theory and laboratory parameters are input into MAGIC in terms of PIC parameters. The PIC parameters of primary interest to the theory are: 1) beam emission area, 2) beam energy parameters, and 3) beam current density parameters such as beam current density as a function of time. For example, beam current density as a function of time sets the general shape of the temporal pulse train. This shape determines the amplitude, depth of modulation, and modulation frequency for the pure sine-squared type of pulses used in the MAGIC simulations. I will present the conversion equations relating laboratory and PIC parameters when I develop my analytic comparison of the theory to the PIC data. For now, I will use PIC parameters because these are the parameters actually used in the PIC simulations to generate data. For 1-D PIC simulations, it is possible to explore the field quantities, particle dynamics, and generated radiation with respect to modulation depth D, emitted electron energy eV_b, average electron beam current density $\langle j \rangle$, and beam current density modulation frequency ν_{mod}. Of course, it is understood that these parameters are given in terms of all the previously defined common features and settings. Note that the beam emission area A could be simulated, but only for various small spots and never (without resorting to PIC code modifications and simulation tricks or physical scaling) for real 2-D spots where significant diffraction effects show up. So area scaling among small spots would be uninteresting, but doable. The lobe pattern of the radiation with respect to far-field detector angle θ from the 1-D emitted current line is not generated by MAGIC but by a separate postprocessor program, and is not an independent input variable into MAGIC. Instead of trying to model an arbitrary reproduction of a real pulse or an infinite number of pulse envelopes, a simple $\sin^2$ type of modulated pulse train was simulated. Using a simple $\sin^2$ type of modulated pulse train *maximized the useful information obtained from a limited number of simulations because it decreased the number of independent parameters and enabled better comparison with theory predictions.* Although D could have been varied as well for the $\sin^2$ type of modulated pulse train, D was kept at 100% for maximum radiation and was not varied due to limited number of simulations that could be performed. Consequently, for the six PIC parameters discussed above, θ was a dependent variable, A and D were not explored, and eV_b, $\langle j \rangle$, and ν_{mod} were varied.

Results and Graphs

Getting results from the MAGIC code was done in several ways, and the first results to be presented come from graphs that MAGIC generates for the user. These graphs represent physical quantities at a fixed time step (such as particle plots of phase space, or field components) or quantities as they change in time (such as field components and currents). Figures 1-6 were generated in this way by MAGIC to show quantities of interest in the simulation grid. Note that these MAGIC graphs do not employ the theory notation used by the rest of this paper and are instead labeled differently. Consequently, x and y on the MAGIC graphs represent z and x respectively, as used in this paper.

Figures 1-3 are particle phase space graphs showing particle p_z momenta per unit mass (left graph) and particle x-coordinates (right graph) versus particle z-coordinates as snapshots at a given PIC simulation time. The left graphs in Figs. 1-3 are time snapshots of the electron velocities multiplied by the relativistic factor γ, such that $p_z = \gamma v_z$, and where the z-axis represents the perpendicular distance off the conducting plane used to emit the electrons. The right graphs in Figs. 1-3 are the corresponding time snapshots of the electron positions in the PIC simulation grid, plotted as x versus z coordinates. These particle trajectory graphs show emitted beam electrons frozen in time during their various orbits and the distribution and approximate maximum 'orbit' distance of the steady-state electron trajectories.

Figure 1 particle plots show time snapshots from the early history of an emitted pulse. When a pulse is emitted, the initial electrons escape thc conducting surface and essentially never return. Figure 1 shows the PIC simulation before steady-state, saturated behavior has been achieved. However, the electrons that were initially emitted have already left the PIC simulation grid. Note that many electrons, in returning to the conducting surface from which they were emitted, are starting to approach orbits that are typical of steady state. Also, the simulated electrons have stopped escaping the simulation's right boundary, and the remaining electrons that did not escape that boundary are falling back to the conducting surface from their larger, early simulation time orbits.

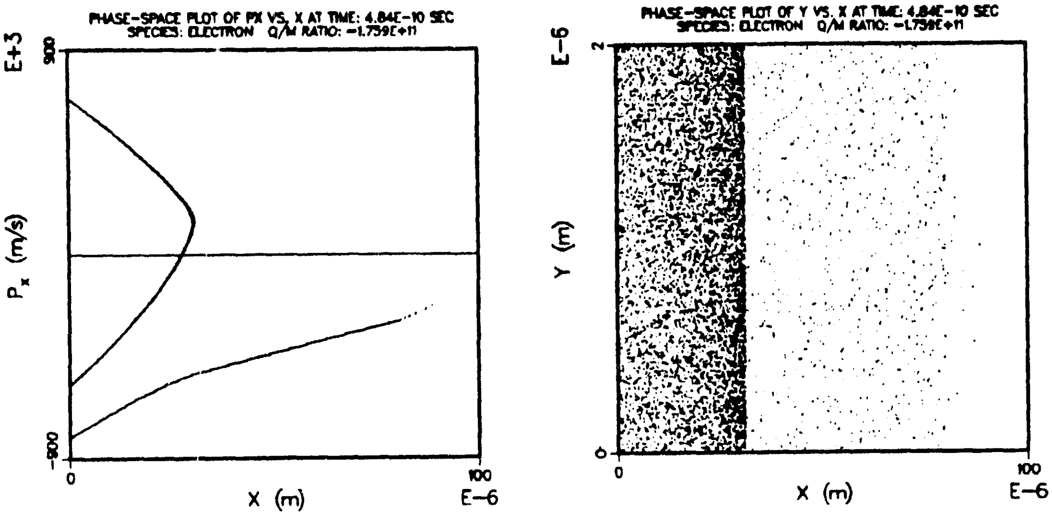

Figure 1: Time snapshots of PIC simulation particle momenta and positions before steady-state, saturated behavior has been achieved. Emitted beam parameters are $eV_b = 1.3305$ eV and $\langle j \rangle = 1722.0$ A/m^2 unmodulated.

Figure 2 particle plots show time snapshots after steady state, or saturation, has been achieved. The left phase space graph essentially shows the electrons being emitted at a given positive velocity off the surface of the conducting plate, reaching a maximum normal orbit distance $d_{\max}$ with zero velocity, and returning to the conducting surface with the emitted velocity in the negative direction. This is a smooth curve, but when the simulation has reached saturation and the space charge is large enough, self-oscillation becomes pronounced for the emitted electrons, and self-modulated radiation is produced at a characteristic or natural frequency I will refer to as ν_{nat}.

The results from my theory paper [Ref. 1] indicate that ν_{nat}, obtained from the processed PIC data as the primary frequency component of the self-modulated radiation's far-field E_Z component, is equal to the self-oscillation frequency of the emitted electrons. Consequently, ν_{nat} is proportional to

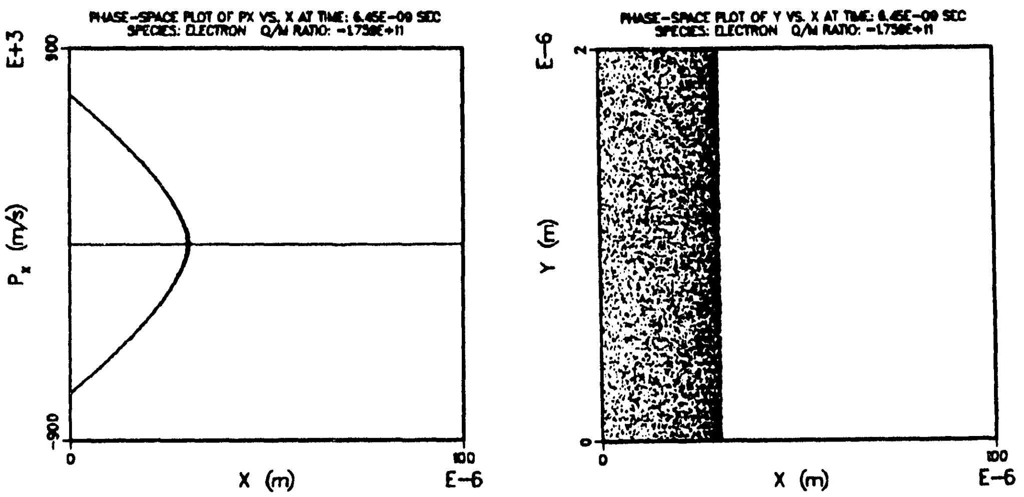

Figure 2: Time snapshots of PIC simulation particle momenta and positions during steady state for beam parameters that do not support self-oscillation. Emitted beam parameters are eV_b = 1.3305 eV and $\langle j \rangle$ = 1722.0 A/m^2 unmodulated.

the square root of the average emitted current density $\langle j \rangle$, and inversely proportional to the square root of the mean emitted electron velocity v_e. This can be written as

$$\nu_{\mathrm{nat}} \propto \sqrt{\frac{\langle j \rangle}{v_e}} \tag{1}$$

This self-oscillation is visible in the left graph of Fig. 3 as a non-symmetric curve due to bunching of the emitted electrons.

Figures 4-6 show selected snapshots of E_Z, the electric field strength component perpendicular to the conducting surface, graphed versus the perpendicular distance z from the conducting surface. Note that in the region of the electron sheath near the conducting surface, the field strengths reach tens of kilovolts per meter. Also, the E_Z values on the graphs do not reach zero at large z and the constant E_Z offset is probably caused by the initially emitted electrons that escaped the right boundary of the PIC simulations.

Other results can be user graphed by selecting output data from MAGIC after the simulations are done, and manually inputting the results into a graphics program for display. Figures 7, 8, and 11-15 were created using the Visual Numerics, Inc. POINT AND CLICK graphics software to graph

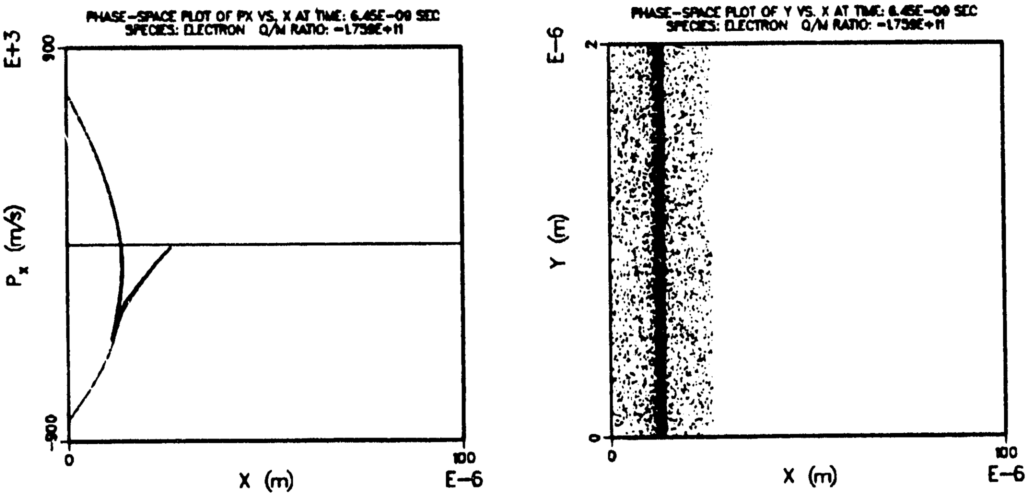

Figure 3: Time snapshots of PIC simulation particle momenta and positions during steady state for beam parameters supporting self-oscillation. Emitted beam parameters are $eV_b = 1.3305$ eV and $\langle j \rangle = 6888.0$ A/m^2 unmodulated.

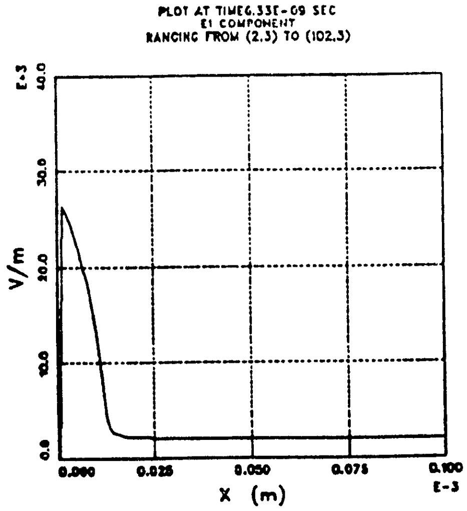

Figure 4: Time snapshot of the E_Z electric field strength component versus perpendicular distance z from the conducting beam emission surface. Emitted beam parameters are $eV_b = 0.2$ eV and $\langle j \rangle = 1150$ A/m^2 unmodulated.

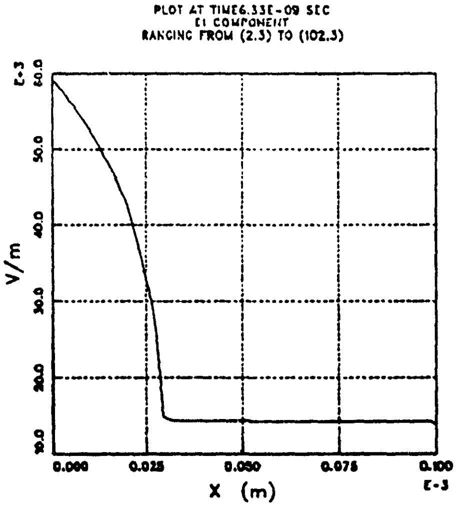

Figure 5: Time snapshot of the E_Z electric field strength component versus perpendicular distance z from the conducting beam emission surface. Emitted beam parameters are eV_b = 1.3305 eV and $\langle j \rangle$ = 1722.0 A/m^2 unmodulated.

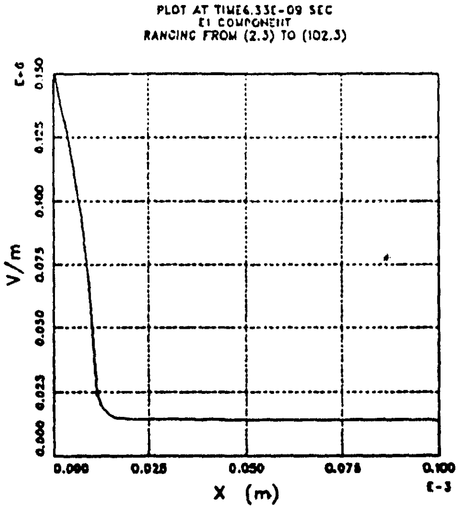

Figure 6: Time snapshot of the E_Z electric field strength component versus perpendicular distance z from the conducting beam emission surface. Emitted beam parameters are eV_b = 1.3305 eV and $\langle j \rangle$ = 15498 A/m^2 unmodulated.

the RFMAGIC.F results generated from various MAGIC runs. Figures 9-10 were graphed using the Los Alamos National Laboratory X-Division's POP graphics program.

Figure 7 is an example where $d_{\max}$, the approximate maximum normal distance of the electron orbits, was estimated from MAGIC trajectory or phase space graphs for various values of current density. The values of $d_{\max}$ were sampled quite coarsely in time (approximately every nanosecond), with the result being that the actual maximum normal electron orbit distances in steady state are estimated somewhat coarsely as well. The sampled data from several constant or unmodulated, monoenergetic, perpendicularly emitted electron current density runs were then plotted to make Fig. 7. Note that the modulated PIC code simulations used the function $j(t) = j_0[1 - \cos(2\pi\nu t)]/2$ to modulate the current density in time at ν, the modulation frequency. The unmodulated PIC code simulations used a constant current density function where $j(t) = j_0/2 = \langle j \rangle$. The modulated PIC code simulations' current density function differs from my theory paper [Ref. 1] current density function where $j(t) = \langle j \rangle - j_0 \cos(2\pi\nu t)$ and $F_D = D \equiv j_0/\langle j \rangle$ are used, but since the PIC code simulations used 100% current density modulation, then my theory paper's j_0 is equivalent to the PIC code's $j_0/2$, and both quantities are equal to $\langle j \rangle$, the average beam emission or photoemission current density. Therefore, for the PIC code simulations and results in this paper, $F_D = D \equiv (j_0/2)/\langle j \rangle = 1$ can be used with the theory paper's equation $j(t) = \langle j \rangle[1 - F_D \cos(2\pi\nu t)]$ for consistency between the two papers. Consequently, for all of the PIC simulations results presented in this paper, $\langle j \rangle$, the unmodulated or average modulated photoemission current density, is related to $j_0/2$, the sinusoidal amplitude of the modulated photoemission current density, by $\langle j \rangle = j_0/2$ at $F_D = D = 100\%$.

However, the rest of the graphs I will present as typical of the results of all the PIC code simulations herein require knowing far-field radiation quantities. This is a problem, since PIC codes generally do not calculate these types of physical quantities outside of the simulation grid, and for this problem, the far-field based on the frequency of the radiation generated is many times larger than the size of the simulation grid. In order to solve this problem, I created a postprocessor named RFMAGIC.F to take the MAGIC time history data of the emitted current and calculate various far-field radiation quantities of interest.

RFMAGIC.F is a FORTRAN program that converts the time history

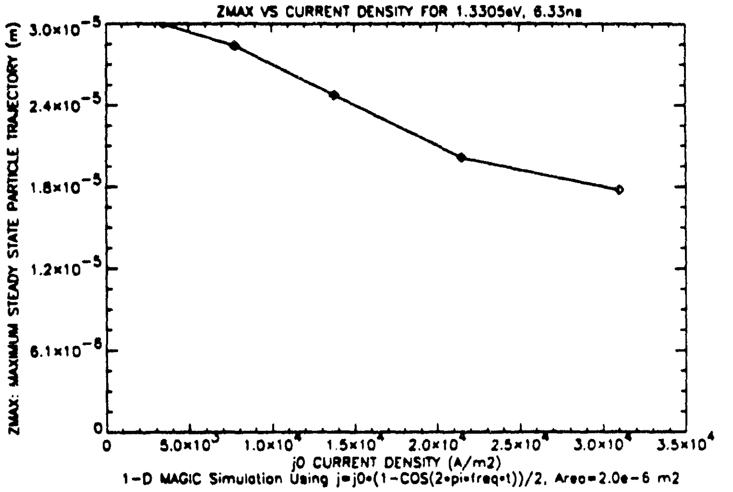

Figure 7: d_{max}, the approximate maximum steady-state orbit distance of unmodulated, monoenergetic electrons that are perpendicularly emitted from a conducting surface, versus j_0, twice the average emitted current density. The emitted electron energy was a constant 1.3305 eV for the graph.

data of a one-dimensional current source into far-field radiation parameters. To do this conversion, MAGIC is first set up to do one-dimensional beam emission simulations that dump the values of the current component in the z-direction with respect to two parameters, z and t. The parameter z represents the normal (perpendicular) distance from the conducting surface in meters, whereas the parameter t represents the simulation dump time step in seconds. This creates a table of current source values that can be represented as $I_z(z,t)$ for the one-dimensional PIC simulations. RFMAGIC.F then reads these current values into a matrix and treats them as a current source for the production of radiation by calculating the time derivatives of the current values and then translating the values of this time derivative current source into radiated far-field electric and magnetic field components. The postprocessing code accomplishes these field component calculations by fully implementing the necessary one-dimensional current source forms of the equations listed in my theory paper at the start of the subsection titled *Radiation Equations* [Ref. 1]. As such, full retarded time effects are included in the calculations. Also, in addition to the one-dimensional cylindrical current source, a symmetric two-dimensional slice of a hemisphere having a fixed, far-field radius is used for the observer calculations. Once the radiated field components are known for each observer point in time and space, it is easy to calculate the Poynting vector and $d\langle P\rangle/d\Omega$ for various θ's, the average power radiated through a forward hemisphere (on the emission side of the conducting plane, without reflections) versus time, $\langle P_r\rangle_h$, and the integrated or time-independent radiated energy through a forward hemisphere in the far-field, E_{rh}.

A general overview of the RFMAGIC.F code highlighting key features and numerical techniques is listed below:

1. Reads and stores the 1-D current source data values of $I_z(z,t)$ that were previously dumped, periodically in simulation (source) time, from MAGIC's simulation grid.

2. Replaces the current source $I_z(z,t)$ matrix values with their corresponding time derivative values of the current $\dot{I}_z(z,t)$, using three-point difference formulaes.

3. Calculates, for each observer time step, necessary radiated electric and magnetic field components in the observer's space and time frame using the PIC $\dot{I}_z(z,t)$ data from the source's space and time frame.

    (a) The radiated far-field electric and magnetic field components are calculated in the observer's frame at regularly spaced polar angles (in θ) over a symmetric slice of hemispherical shell having a fixed, far-field radius.

    (b) RFMAGIC.F spatially integrates the $\dot{I}_z(z,t)$ values over z in the source's frame *at retarded source times*, performing the integration for every radiated field component calculated at every point in space (and time) in the observer's frame.

        i. Integrates over the $\dot{I}_z(z,t)$ source coordinates using five-point, multistep Gaussian quadrature.

        ii. Uses bilinear interpolation to calculate the corresponding value of the $\dot{I}_z(z,t)$ source for each required integration z and retarded time t coordinate.

4. Calculates, for each observer time step, a variety of far-field radiation parameters in the observer's frame using the calculated radiated far-field electric and magnetic field components.

(a) Calculates the Poynting vector **P** and the instantaneous power per unit solid angle $dP/d\Omega$ at the regularly spaced polar angles.

(b) Integrates $dP/d\Omega$ over the polar angles to calculate P_{rh}, the total instantaneous power radiated through a far-field hemisphere.

5. Evolves the far-field radiation parameters over time in the observer's frame using all of the far-field contributions from the $\dot{I}_z(z,t)$ PIC code simulation data to generate the PIC code simulation's far-field radiation through a forward hemisphere.

   (a) Generates quantities versus time such as $P_{\text{rh}}(t)$.

   (b) Generates time averaged quantities such as $d\langle P\rangle/d\Omega$ and $\langle P_r\rangle_h$.

   (c) Generates time integrated quantities such as radiated energy E_{rh}.

Basically two sets of PIC data exist, corresponding to two values of emitted electron energy, eV_b. PIC simulations at 0.2 and 1.3305 eV were conducted. At 0.2 eV, a resonance curve was generated. At 1.3305 eV, a more detailed resonance curve was constructed, as well as several other types of graphs showing far-field radiation results. Figures 8-14 correspond to MAGIC runs with an emitted electron energy of 1.3305 eV. Figure 15 corresponds to MAGIC runs using an emitted electron energy of 0.2 eV.

Figure 8 was generated from several MAGIC simulations to show the variation of natural frequency ν_{nat} versus twice the average emitted electron beam current density $\langle j\rangle$. This natural frequency curve was generated by graphing the primary FFT frequency component of the far-field radiation's $E_Z(t)$ electric field component, as measured in the far-field at an observer angle of 80 degrees in θ, and where the radiation was generated from the self-oscillation of the emitted electrons. The natural frequency was simulated by emitting a perpendicular, constant or unmodulated, monoenergetic electron beam off of a conducting plane. For the electron energy and current values used, sufficient space charge was generated to initiate electron bunching and self-oscillation. However, in steady state, the j_0=3444.0 A/m^2 simulation's self-oscillation damped out. By modulating the emitted electron beam current density $\langle j\rangle$ at its natural or self-oscillation frequency so $\nu_{\text{mod}} = \nu_{\text{nat}}$, while using a sine-squared type of pulse train with the same average current density as the self-oscillating simulation, maximum radiated power was achieved. This maximum radiated power, or *resonance power*, occurs when the modulation frequency resonates with the natural frequency.

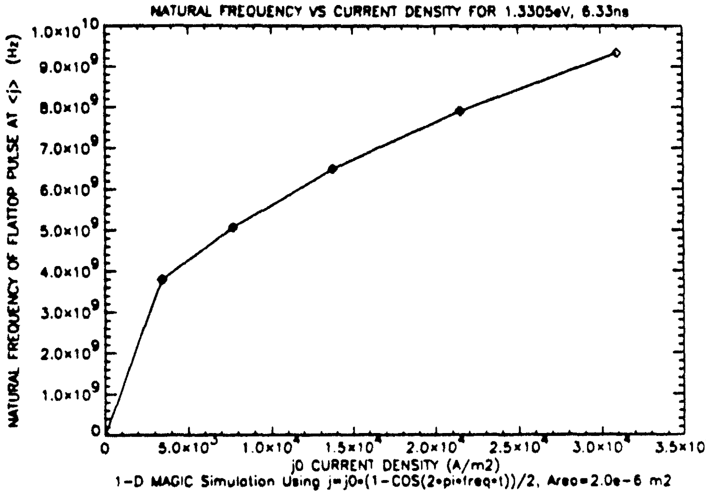

Figure 8: ν_{nat}, the primary FFT frequency component of the 'natural' or self-oscillation generated far-field radiation that spontaneously occurs when an unmodulated, monoenergetic electron beam is perpendicularly emitted from a conducting surface, versus twice the average emitted current density. The emitted electron energy was a constant 1.3305 eV for the graph.

Figures 9-10 are graphs made from two MAGIC simulations and subsequent RFMAGIC.F postprocessing that show P_{rh}, the instantaneous power radiated through a far-field hemisphere, versus time (left graph) and E_{rh}, the corresponding integrated energy radiated through a far-field hemisphere, versus time (right graph). Figure 9 shows the case used in Figs. 1, 2, 5, 7 and 8 where the self-oscillation damps out in steady state. The primary radiation generated, in this case, seems to come from electrons emitted early in the simulation. The PIC code itself could be contributing to damping out the self-oscillation in steady state due to its algorithms, but this possibility was not investigated for the purposes of this paper. Figure 10 shows a case *typical* of all the rest of the MAGIC simulations where self-oscillation radiation is measured, where though the electrons emitted early on form a sharp power pulse, the generated radiation is soon dominated by the steady-state self-oscillation radiation. The spiky nature of the P_{rh} versus time curve is characteristic of all self-oscillating or modulated PIC simulations. When

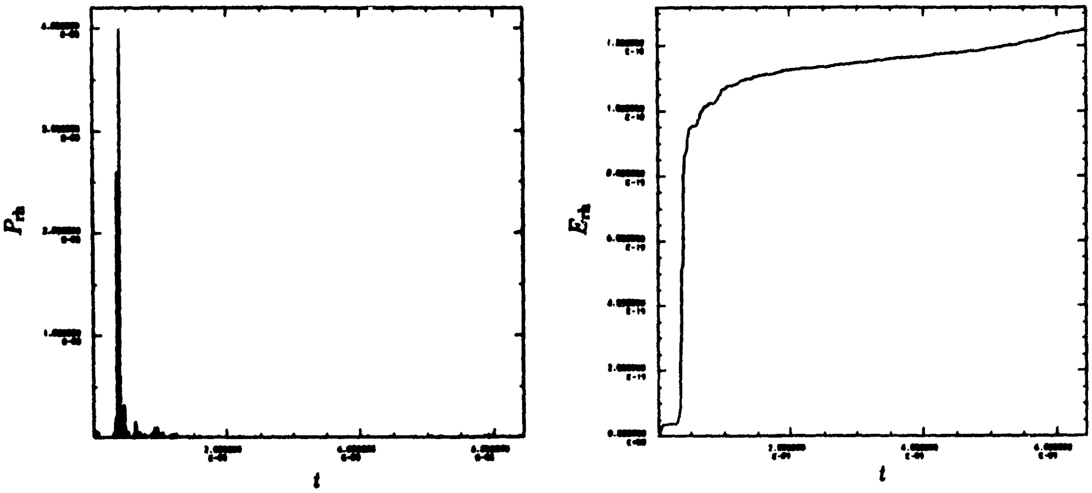

Figure 9: Far-field radiated power $P_{\rm rh}$ and cumulative energy $E_{\rm rh}$ versus time for a MAGIC simulation that does not display significant self-oscillation in steady state. Emitted beam parameters are eV_b = 1.3305 eV and $\langle j \rangle$ = 1722 A/m^2 unmodulated.

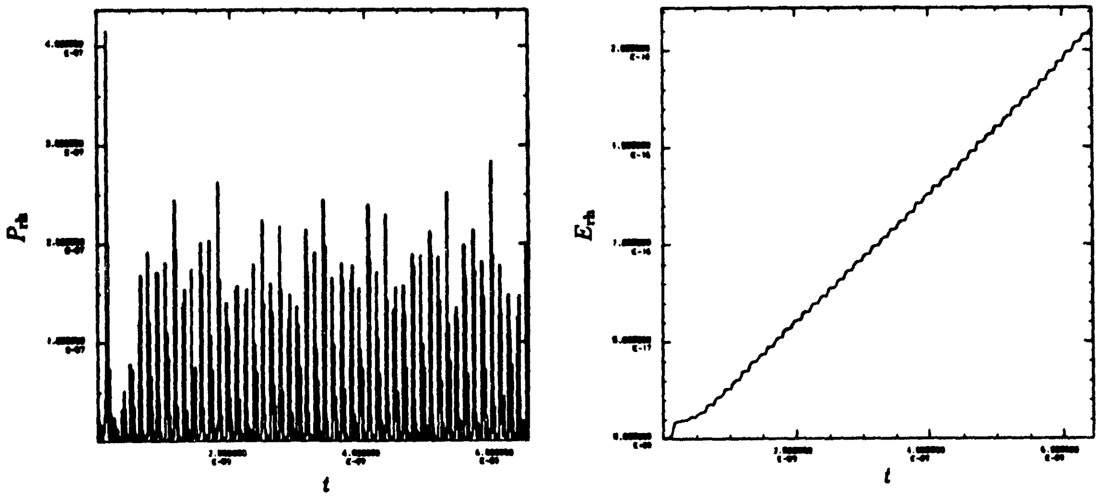

Figure 10: Far-field radiated power $P_{\rm rh}$ and cumulative energy $E_{\rm rh}$ versus time for a MAGIC simulation that achieves self-oscillation in steady state. The natural frequency of the self-oscillation generated radiation is $\nu_{\rm nat}$ = 7.9074 GHz with a 0.15815 GHz FFT box width. Emitted beam parameters are eV_b = 1.3305 eV and $\langle j \rangle$ = 10757.5 A/m^2 unmodulated.

$P_{\mathrm{rh}}(t)$ is integrated over time for PIC simulations with self-oscillation *or* ν_{mod} modulated current density, then *typically*, as seen in Fig. 10, the cumulative E_{rh} radiated energy versus time curve increases in a very linear manner, with only small oscillations about a best fit line through the curve.

Figure 11 shows the typical far-field lobe pattern of the calculated radiation. It is a graph of $d\langle P\rangle/d\Omega$ versus θ for a single MAGIC run and is typical of the $\sin^2\theta$ pattern seen for all of the MAGIC runs that were sampled for this pattern. Graphing $d\langle P\rangle/d\Omega$ versus θ is an excellent test of both the small-spot PIC code simulations and the RFMAGIC.F code since the pattern of the far-field radiation lobes based on the MAGIC simulation data set *and* its postprocessing by RFMAGIC.F *reproduces the* $\sin^2\theta$*, or classic dipole radiation pattern, expected for a small-spot* [Ref. 1]. The dotted line cons:sts of the actual data points RFMAGIC.F dumps. To show how close the data points are to a $\sin^2\theta$ curve, the continuous line is a $\sin^2\theta$ curve that was normalized to the maximum value of the RFMAGIC.F data. The agreement is very close even without curve fitting the normalization coefficient of the $\sin^2\theta$ equation to all of the PIC data.

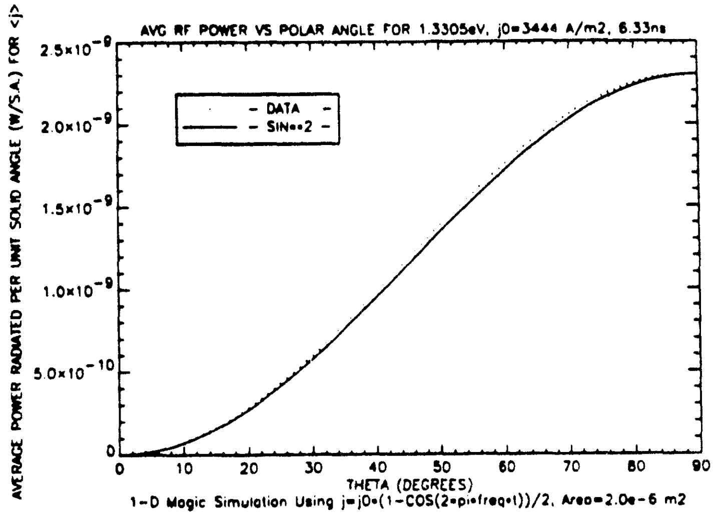

Figure 11: $d\langle P\rangle/d\Omega$ versus θ characteristic small-spot far-field radiation lobe pattern *typical* of 1-D PIC simulations. The y-axis should be divided by 44.32 to scale $d\langle P\rangle/d\Omega$ correctly. Emitted beam parameters are $eV_b = 1.3305$ eV and $\langle j\rangle = 1722.0$ A/m^2 unmodulated.

Figure 12 is a resonance curve (F_R) generated from 19 MAGIC simulations. It shows how modulating the emitted beam current density at ν_{mod} frequencies other than the natural or resonance frequency decreases the total E_{rh} energy radiated through a hemisphere. The natural frequency of the self-oscillation generated radiation for a constant, injected beam having the same eV_b and $\langle j \rangle$ as Fig. 12 is $\nu_{\mathrm{nat}} = 6.0096$ GHz, with a 0.15815 GHz FFT box width. The modulation frequency ν_{mod} was simulated over zero (actually an unmodulated, constant beam emitted at $\langle j \rangle$) to 18 GHz sampled at every gigahertz. As such, the graph spans 0.0 to 3.0 times the natural frequency. Note that besides the main peak, another lesser maximum is occurring near 18 GHz. Often, a resonance plot is modeled by a Lorentzian curve, but this resonance curve's behavior is more complicated, and even looks Gaussian near the resonance peak. However, for behavior near the resonance global maximum, which is an interesting region because it maximizes radiation with respect to ν_{mod}, a well chosen Lorentzian would suffice to model the *near-resonance* behavior.

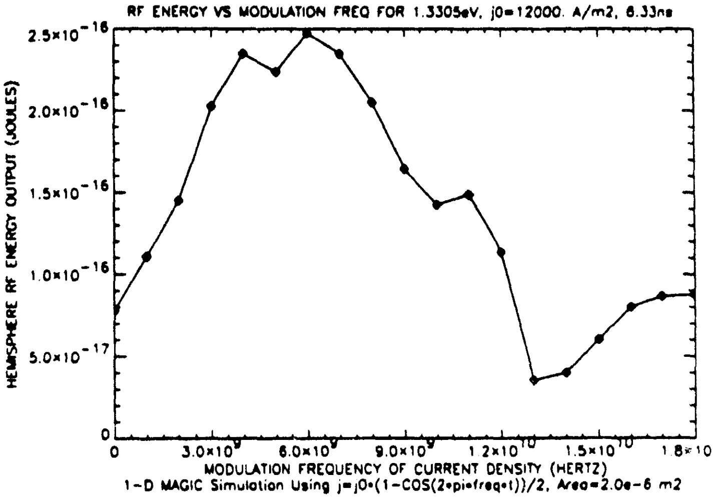

Figure 12: F_R resonance curve of small-spot, far-field radiation for emitted beam parameters $eV_b = 1.3305$ eV and $\langle j \rangle = 6000$ A/m^2. Graph shows E_{rh}, the total far-field energy radiated through a hemisphere, verses ν_{mod}, the modulation frequency of the emitted current density.

Figures 13 and 14 are similar. Figure 13 shows E_{rh}, the total energy radiated through a hemisphere in the far field, versus $E_B = \langle P_B \rangle t_{\mathrm{run}}$, the total input beam emission energy, for various $t_{\mathrm{run}} = 6.3263$ ns PIC code simulated

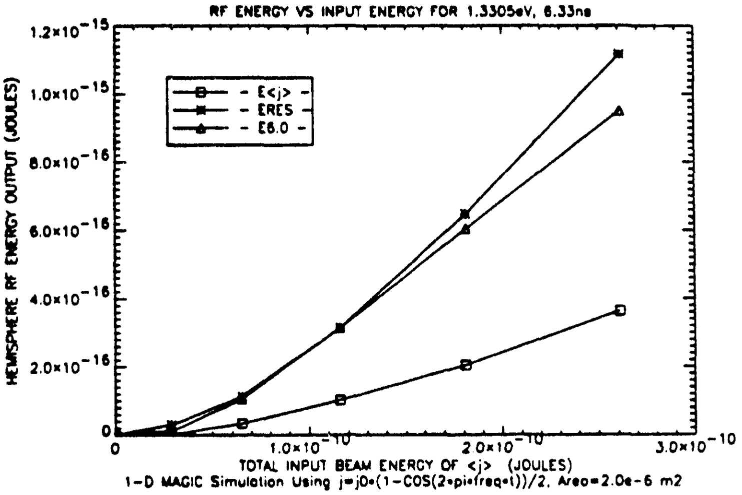

Figure 13: E_{rh}, the total far-field energy radiated through a hemisphere, versus E_B, the total input beam emission energy, for several ν_{mod} strategies. Emitted electron energy was a constant 1.3305 eV for the curves.

beam emission pulse trains. Figure 14 uses the same data as Fig. 13, but shows the radiation conversion efficiency η_{rhB} versus j_0, where j_0 for the PIC code simulations is equal to $2\langle j\rangle$, twice the average emitted current density. The conversion efficiency η_{rhB} is basically the total far-field energy radiated through a hemisphere, calculated using the emitted electrons in free space (with the conducting emission plane removed), divided by the total input beam emission energy. So η_{rhB} represents about half of the actual conversion efficiency radiated through a sphere, with or without the conducting plane. This halving of the generated radiation occurs because P_{rh} and E_{rh} were calculated by RFMAGIC.F as the power and energy radiated through the *forward hemisphere* of the simulated problem with the conducting plane removed and the radiation calculations performed for the electron sheath in free space. This RFMAGIC.F calculated radiation can be compared with the radiation for the problem being simulated because for small spots having small photoemission orbit distances above the conducting plane, where larger radiation wavelengths are involved, the forward and conducting plane reflected radiation wavelengths generally interfere constructively from what is essentially a point source radiator. So while RFMAGIC.F only integrated $dP/d\Omega$ over a symmetric forward hemisphere to calculate P_{rh} in free space

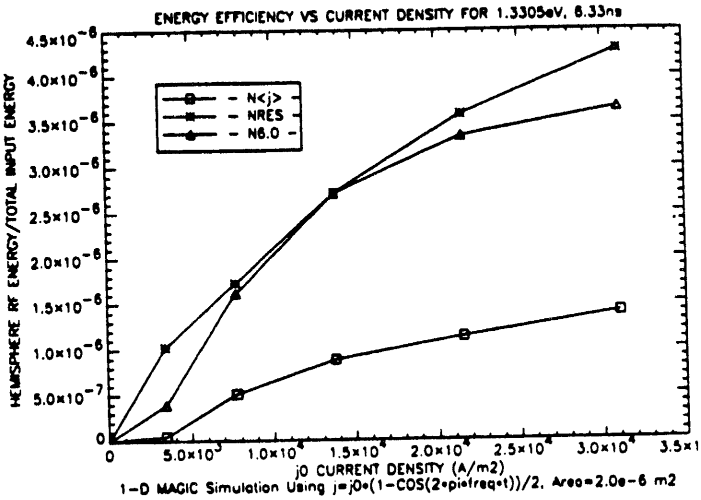

Figure 14: η_{rhB}, the conversion efficiency of the total input beam emission energy to total far-field energy radiated through a hemisphere, versus twice the average emitted current density for several ν_{mod} strategies. Emitted electron energy was a constant 1.3305 eV for the curves.

without the conducting plane, this result is equivalent to calculating the radiation simulated from beam emission off of a conducting plane that restricts the radiation to essentially the forward hemisphere, and where the reflected radiation off of the conducting plane (approximately half of the total radiation) was not included in the RFMAGIC.F calculations.

η_{rhB} and several other useful efficiency and power equations are listed below.

$$\langle P_r \rangle_h = E_{\mathrm{rh}}/t_{\mathrm{run}} \tag{2}$$

$$\langle P_B \rangle = |e|eV_b \frac{A\langle j \rangle}{e} \tag{3}$$

$$\langle P_L \rangle = \frac{hc}{e} \frac{A\langle j \rangle}{\lambda_L Y} \tag{4}$$

$$\eta_{\mathrm{rhB}} = \frac{\langle P_r \rangle_h}{\langle P_B \rangle} = \frac{1}{|e|eV_b} \frac{e}{A\langle j \rangle} \frac{E_{\mathrm{rh}}}{t_{\mathrm{run}}} \tag{5}$$

$$\eta_{\mathrm{BL}} = \frac{\langle P_B \rangle}{\langle P_L \rangle} = |e|eV_b \frac{\lambda_L Y}{hc} \tag{6}$$

$$\eta_{\rm rhL} = \frac{\langle P_r\rangle_h}{\langle P_L\rangle} = \frac{e}{hc}\frac{\lambda_L Y E_{\rm rh}}{A\langle j\rangle t_{\rm run}} \tag{7}$$

$$\langle P_r\rangle_s \approx 2\langle P_r\rangle_h \tag{8}$$

$$\eta_{\rm rsB} \approx 2\eta_{\rm rhB} \tag{9}$$

$$\eta_{\rm rsL} \approx 2\eta_{\rm rhL} \tag{10}$$

The above equations show how $\eta_{\rm rhB}$ was calculated for Fig. 14 and provide convenient expressions for converting the radiated energy simulated by MAGIC, using a beam emission model, into other useful quantities. For example, $\eta_{\rm rsL}$ is an important quantity characteristic of a laboratory experiment that uses a photoemission light source, that was not directly simulated computationally by the beam emission model used, but which can be estimated by using the listed equations. The parameter $\langle P_B\rangle$ is the average power of the emitted beam of simulated photoelectrons, while $\langle P_L\rangle$ is the average power of the light source measured over the photoem'ssion spot area on the conducting surface. Most of the parameters are defined in a later subsection titled *Curve-Fitting Equations*, and the MAGIC and RFMAGIC.F postprocessed data used in this paper are listed in **Appendix A**. By picking appropriate values of λ_L and Y, the PIC code simulation data can be converted into various experimental predictions for $\langle P_L\rangle$ and total input light to output radiation through a sphere conversion efficiency $\eta_{\rm rsL}$, as well as the intermediate input light to output electron beam energy conversion efficiency $\eta_{\rm BL}$. Note that the elementary charge e in SI units, when written as $|e|$ in this paper, becomes a conversion number having units of Joules per eV. So $|e|$ is a number having the magnitude of e in SI that is multiplied with quantities having units of eV to convert those quantities to values having dimensions of Joules. Consequently, the parameter eV_b, which represents the energy per emitted electron in units of eV, is always associated with $|e|$ in the equations. The units of eV are *always* used with eV_b and later on with the work function of the photoemission surface ϕ, because they are more convenient numbers to work with and for looking up the published ϕ values, which are usually listed in units of eV. Otherwise, the equations and parameters in this paper are generally applied using SI units.

Each graph in Figs. 13-14 is divided into three curves. Each curve is constructed using data obtained from five MAGIC simulations. The curves show the effect of not modulating the electron beam $(E_{\langle j\rangle})$, of modulating the beam at a constant 6.0 GHz $(E_{6.0})$, and of modulating the beam at the natural frequency for maximum resonance radiated energy (E_{RES}). For the emitted electron energy and average current density used, the natural frequency varies from approximately 3.2 to 9.3 GHz as the current density varies from its lowest to highest value. Note that the lowest curve is the unmodulated, or self-oscillating curve, and that the highest curve is the curve where the modulation frequency is increased from about 3.2 to 9.3 GHz to maintain resonance as the average current density increases. As such, the beam which is always modulated at 6.0 GHz peaks up to the resonance curve's value near the middle of the graph when the beam parameters also reach a natural frequency of close to 6.0 GHz. Note that the resonance radiated energy and conversion efficiency values are at most only three to four times greater than the unmodulated beam emission values for radiated energy and efficiency. Also, for modulation frequencies less than the resonance frequency, the decreases in radiated energy did not drop below the unmodulated beam emission values. However, Fig. 12 shows it is possible that over-modulation, or using frequencies greater than the resonance frequency, can decrease the amount of radiated energy to values that are even smaller than the unmodulated beam emission values.

The Fig. 15 E_{rh} versus ν_{mod} curve is similar to the Fig. 12 E_{rh} versus ν_{mod} curve, except that it was simulated with an emitted electron energy of 0.2 eV, is less well resolved, and only spans 0.0 to 2.5 times the natural frequency. The natural frequency of the self-oscillation generated radiation for Fig. 15 is $\nu_{\mathrm{nat}} = 4.2700$ GHz with a 0.15815 GHz FFT box width. Note that the lesser number of points on Fig. 15 do not capture the shape of F_R, the steady-state, small-spot, resonance curve, as well as Fig. 12. More data were generated later that were not included in Fig. 15. **Appendix A** lists tables showing *all* of the F_R curves' E_{rh} data generated from the PIC code simulations. Still, Fig. 15 shows that the shape of the near-resonance maximum for 0.2 eV emitted electrons is similar to the shape of the Fig. 12 near-resonance maximum for 1.3305 eV emitted electrons. The major difference between the two curves is that the E_{rh} radiated energy is less for Fig. 15, as theory predicts [Ref. 1] for lower emitted electron energy and lower average current density. Together, the two resonance curves provide a variation of two levels of emitted electron energy in an interesting region of parameter space.

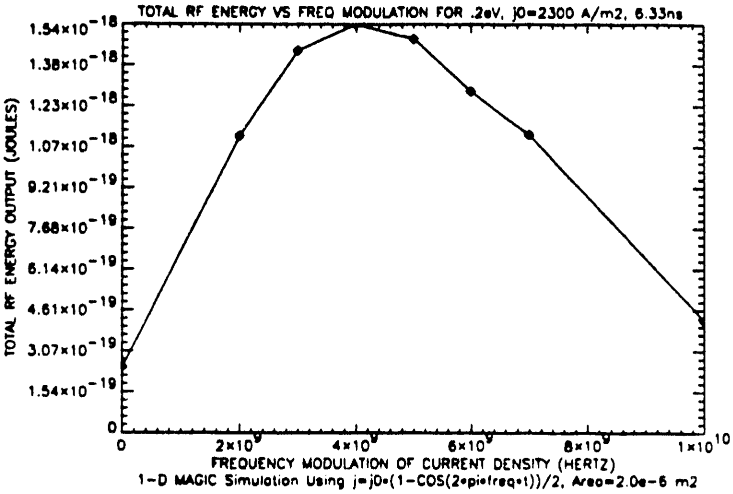

Figure 15: F_R resonance curve of small-spot, far-field radiation for emitted beam parameters $eV_b = 0.2$ eV and $\langle j \rangle = 1150$ A/m^2. Graph shows $E_{\rm rh}$, the total far-field energy radiated through a hemisphere, verses $\nu_{\rm mod}$, the modulation frequency of the emitted current density.

Comparing Theory and PIC Results

This section will 'analytically' compare the PIC data with the theory equations presented in my theory paper [Ref. 1]. The graphs presented in the last section show some of the typical PIC code simulation parameters and how they behave. While these qualitative graphs are informative, the PIC data can also be used to get more quantitative, analytic type results that can readily be compared with the theory. The more quantitative, analytic type results and comparisons are accomplished by means of curve fitting. Generally, this paper will present the basic curve-fitting topics needed to understand the comparisons of the theory [Ref. 1] and PIC code simulation results made in this paper. If more information is desired, many excellent books exist that explain the basics of curve fitting and data analysis [Ref. 4],[Ref. 5]. First I will list the 'generalized' PIC parameter set theory equations with unknown primed constant coefficients and exponents. Next I will discuss the rationale behind selecting curve fitting to analyze the PIC code data and why it is so useful to test theory equations versus data. Then I will explain the actual concepts and the process involved in performing the curve fitting. Finally, I will present and analyze the nonlinear curve-fitting results of the generalized PIC parameter set theory constants.

Curve-Fitting Equations

The approximate steady-state theory equations for maximum electric field strength component $E_{Z\ \mathrm{max}}$, maximum electron orbit distance d_{max}, natural or resonance frequency ν_{nat}, and average small-spot, far-field power radiated through a sphere $\langle P_r\rangle_s$, can be expressed as generalized theory equations, or *Generalized Scaling Laws* [Ref. 1], as:

$$
\begin{aligned}
E_{Z\ \mathrm{max}} &= a'\beta^{b'}\langle j\rangle^{c'} && (11)\\
d_{\mathrm{max}} &= a'\beta^{b'}\langle j\rangle^{c'} && (12)\\
\nu_{\mathrm{nat}} &= a'\beta^{b'}\langle j\rangle^{c'} && (13)\\
\langle P_r\rangle_s &= a'F_D^{d'}A^{e'}\beta^{b'}\langle j\rangle^{c'}F_R && (14)
\end{aligned}
$$

The parameters used in the above equations have the following definitions. $F_D = D$ in this paper, where D is the modulation depth of the sinusoidally modulated beam emission or photoemission current density pulse train versus time having modulation frequency ν_{mod}. For the PIC simulations performed in this paper, $D = 1$ was used for 100% sinusoidally modulated electron current density emission. A is the electron emission area on the conducting surface and was set constant at $A = 2.0 \times 10^{-6}$ m^2 for all of the PIC simulations. β represents the emitted electron velocity divided by the speed of light in a vacuum. The parameter $\langle j \rangle$ is the average emitted current density. ν_{nat} is the natural or self-oscillation primary frequency component that spontaneously arises when an unmodulated, constant electron beam having current density $\langle j \rangle$ and emitted electron energy eV_b is injected, perpendicular to the conducting surface, into the simulation. F_R is an arbitrary term representing a resonance function that is multiplied with the resonance $\langle P_r \rangle_s$ theory equation to create an approximate nonresonance $\langle P_r \rangle_s$ equation that allows near-resonance power data to be analyzed. Finally, associated equations and units of most parameters in this paper are in SI. However, eV_b and ϕ are exceptions, as they always have units of eV associated with them.

The above equations represent the theory scaling laws with primed unknown constant coefficients and exponents, using the PIC parameter set. Calculation of the unknown generalized theory constants from the PIC data is best accomplished by using equations that contain the parameters actually used by the PIC simulations, i.e., the PIC parameter set. However, to see how the fitted constants propagate into the same equations when written in terms of photon parameters requires applying the following two conversion relations,

$$\beta_{\mathrm{class}} = \sqrt{\frac{2e}{m_e c^2}}\sqrt{\frac{b_1}{\lambda_L} - \phi_s} \qquad (15)$$

$$\langle j \rangle = \frac{1}{b_1}\frac{\lambda_L Y \langle P_L \rangle}{A} \quad \text{where} \qquad (16)$$

$$b_1 \equiv \frac{hc}{e} \quad \text{and} \quad \phi_s \equiv \frac{|e|\phi}{e} \qquad (17)$$

to get the corresponding generalized theory equations converted to my photon parameter set, which I also refer to as my ultraviolet (UV) or laboratory parameter set,

$$E_{Z\,\mathrm{max}} = a'k_1\left(\frac{b_1}{\lambda_L} - \phi_s\right)^{b'/2}\left(\frac{\lambda_L Y \langle P_L \rangle}{A}\right)^{c'} \qquad (18)$$

Curve-Fitting Equations

$$d_{\mathrm{max}} = a'k_1 \left(\frac{b_1}{\lambda_L} - \phi_s\right)^{b'/2} \left(\frac{\lambda_L Y \langle P_L \rangle}{A}\right)^{c'} \tag{19}$$

$$\nu_{\mathrm{nat}} = a'k_1 \left(\frac{b_1}{\lambda_L} - \phi_s\right)^{b'/2} \left(\frac{\lambda_L Y \langle P_L \rangle}{A}\right)^{c'} \tag{20}$$

$$\langle P_r \rangle_s = a'k_1 F_D^{d'} \left(\frac{b_1}{\lambda_L} - \phi_s\right)^{b'/2} \lambda_L^{c'} Y^{c'} A^{e'-c'} \langle P_L \rangle^{c'} F_R \tag{21}$$

$$k_1 \equiv \left(\frac{2^{b'} e^{b'+2c'}}{m_e^{b'} h^{2c'} c^{2(b'+c')}}\right)^{1/2} \tag{22}$$

where e is the elementary charge, m_e is the electron mass, h is the Planck constant, and c is the speed of light in vacuum.

From the conversion relations, since ϕ is the value of the conductor's work function having units of eV, the reader can easily look up the value of ϕ in books, which generally list the work function in eV. Also, ϕ_s is an SI quantity having the same magnitude as ϕ, but with dimensions of volts, which makes it easy to input a number for ϕ_s into the generalized theory equations. Finally, since ϕ and eV_b are parameters that always have units of eV, then the emitted electron energy eV_b, in eV, is calculated from ϕ as

$$eV_b = \frac{hc}{|e|\lambda_L} - \phi \tag{23}$$

Note that the parameters in the photon parameter set cannot all be fitted independently in every parameter when using PIC code *beam emission simulation data* as the beam emission data are described in terms of the PIC parameter set. Consequently, many photon or UV parameters have similar fitted constants because the PIC parameters often represent several photon parameters that scale as a group. For example, there are an infinite number of values for the wavelength of the light source λ_L, the photoelectric yield Y, the average light source power $\langle P_L \rangle$, and photoelectron emission area A that can represent a single, unique value of the average emitted current density $\langle j \rangle$. Similarly, there are an infinite number of values for λ_L and for the work function of the conducting surface ϕ that can represent a single, unique value of the classical expression for β. As a benefit, by using the generalized theory equations and the conversion relations, scanning a few PIC parameters is an efficient way to search many UV parameters. So using a parameter set developed to model PIC simulations that used *a beam emission model has the* *virtue of fewer parameters to search, measure and fit.* Of course, the PIC-to-UV conversion relations must realistically capture the basic physics being modeled. If laboratory experiments are conducted that measure variations and take data using UV parameters, then a generalized UV parameter equation can be fitted. However, even then one would hope that the PIC-to-UV converted parameters would be similarly scaled with respect to each other in their fitting constants, so that the smaller PIC parameter space to fit could itself act as additional information with which to constrain the fit. Since using as much information as possible from theoretical, computational, and laboratory research is desirable in any fitting process, one should first try fitting the constants for the generalized PIC parameter set theory equation, *even with laboratory experimental data measured in UV or other parameter sets.* Besides, the theory equations are simpler to work with and easier to understand when using the PIC parameter set.

What are the values of these generalized constants (coefficients and exponents)? Table 1 shows the values for the constants based on the theory results from my theory paper [Ref. 1].

Table 1: Theoretical Values of Coefficients and Exponents

Eqn.	$a_1 = a'$	$a_2 = d'$	$a_3 = e'$	$a_4 = b'$	$a_5 = c'$
$E_{Z\ \mathrm{max}}$	13875	N/A	N/A	0.5	0.5
d_{max}	18.415	N/A	N/A	1.5	-0.5
ν_{nat}	4.0700×10^6	N/A	N/A	-0.5	0.5
$\langle P_r \rangle_s$	24.657	2.0	2.0	2.0	2.0

Later, using the PIC data listed in **Appendix A**, several fits in one or more theory constants will be presented to see how the PIC fitted constants compare with the theory values. Note that the radiated power equations are for small spots (spatially and temporally coherent electron motion and no significant radiation source phase differences). Also, the single-pulse model used was one dimensional with perpendicular, monoenergetic electron beam emission of a single, rigid, and 'virtually' thin charge sheet. However, all the final equations were derived for steady-state resonance results, i.e., for situations where $\nu_{\mathrm{mod}} = \nu_{\mathrm{nat}}$.

The d_{max} data, as mentioned previously, were coarsely obtained by manually measuring graphs dumped by MAGIC approximately every nanosecond.

Curve-Fitting Equations

For each PIC simulation, the phase space or electron trajectory plot with the maximum electron orbit distance was used to approximate d_{max}. The type of MAGIC graphs used to sample the d_{max} data was presented in Figs. 1-3, and the results were summarized in Fig. 7. As such, the data taken are approximate. These data were obtained from PIC simulations having constant, unmodulated beam emission at various $\langle j \rangle$ and β values. However, self-oscillation usually occurred in the simulations, making it more difficult to accurately determine d_{max}. As a result, the sampled data could range anywhere from d_{min} to d_{max}, even though the data selected from the sample graphs favored values closer to d_{max}. Since d_{min} appears to be more robust and and less affected by self-oscillation, future PIC simulations and analysis of d_{min} would provide good data for testing the theory values of b' and c'. Also, higher sampling rates would improve estimating d_{max}.

The $E_{Z\ \mathrm{max}}$ data were coarsely obtained in the same way as d_{max}, from E_Z versus z snapshots taken approximately every nanosecond. The type of MAGIC graphs used to sample the $E_{Z\ \mathrm{max}}$ data was presented in Figs. 4-6. For each PIC simulation, the graph with the largest peak value of E_Z next to the conducting surface was selected to approximate $E_{Z\ \mathrm{max}}$. Again, an unmodulated current density was used, and self-oscillation usually occurred, making it more difficult to accurately determine $E_{Z\ \mathrm{max}}$. Note that $E_{Z\ \mathrm{max}}$ is the maximum E_Z based on *all* of the charge in the simulation at any given time step *plus* the initially emitted charge that escaped the simulation right boundary, so the graphs of $E_{Z\ \mathrm{max}}$ are less variable in their extremes than d_{max}, where small groups of erratic macro particles can more readily increase the sampled d_{max}. In future PIC code simulations, dumping the time history of E_Z would provide easy access to more accurate E_Z data to analyze.

ν_{nat} PIC data were not obtained by locating the ν_{mod} associated with the global maximum of a resonance curve for each $\langle j \rangle$ and β. It was more efficient computationally to simply inject a constant or unmodulated, monoenergetic, perpendicular electron beam at a given $\langle j \rangle$ and β, and use the primary FFT component of the radiation generated from the self-oscillation of the beam to determine the natural frequency of the steady-state electrons. This self-oscillation frequency ν_{nat} was graphed in Fig. 8.

$\langle P_r \rangle_s$ data were calculated using RFMAGIC.F to postprocess the simulated electron sheath obtained from self-oscillating, resonance, and off-resonance MAGIC simulations. Resonance $\langle P_r \rangle_s$ data were obtained from resonance simulations in which $\langle j \rangle$ was modulated at $\nu_{\mathrm{mod}} = \nu_{\mathrm{nat}}$. However, many self-oscillating and off-resonance PIC simulations were performed, resulting in much more nonresonance than resonance radiation data. Figure 10 presented a typical self-oscillation E_{rh} versus time graph, calculated using RFMAGIC.F, while Fig. 13 contains curves of several E_{rh} results versus modulation frequency and input beam energy, where the input beam energy was varied by changing the emitted current density. Figures 12 and 15 also presented curves of E_{rh} versus modulation frequency. Converting the total radiated energy for a given pulse train into average radiated power to analyze was easy due to the linear nature of E_{rh} versus time, which was typical of the linear curve shown in Fig. 10. But if more complicated pulse envelopes and pulse trains had been used, then the radiated power versus time or the total radiated energy E_{rh} would need to be predicted and analyzed for these pulse trains, instead of the average radiated power.

If the resonance radiated power data *and* other PIC simulation radiated power data could all be used together, a better fit might be performed in more parameters. Instead of 5 or 6 points, 41 data points could be used if the nonresonance $\langle P_r \rangle_s$ theory equation were known. For these fits, the nonresonance $\langle P_r \rangle_s$ equation is approximated by the steady-state, 'resonance' $\langle P_r \rangle_s$ theory equation multiplied by F_R, an arbitrarily chosen resonance function. The F_R function chosen has the property that for resonance power data, $F_R = 1$, and the resonance $\langle P_r \rangle_s$ equation is recovered. So at $\nu_{\mathrm{mod}} = \nu_{\mathrm{nat}}$, F_R obtains its maximum or resonance value of one. Also, $0 < F_R \leq 1$, and F_R is equal to $\langle P_r \rangle_s$ at ν_{mod} divided by $\langle P_r \rangle_s$ at ν_{nat}. For nonresonance data, but *near*-resonance data (i.e. $0.0 \leq \nu_{\mathrm{mod}} \leq 2.3\nu_{\mathrm{nat}}$), a Lorentzian-type resonance function is used to represent F_R.

The arbitrary *near*-resonance function is defined as:

$$
\begin{aligned}
F_R &\equiv \frac{a_6^2}{(\nu_{\mathrm{mod}}/\nu_{\mathrm{nat}} - 1)^2 + a_6^2} \quad \text{where} && (24)\\
\nu_{\mathrm{mod}} &= \nu_{\mathrm{nat}} \quad \text{for resonance, and} && (25)\\
\nu_{\mathrm{nat}} &= a_7 \beta^{a_8} \langle j \rangle^{a_9} \;=\; a' \beta^{b'} \langle j \rangle^{c'} && (26)\\
j &= \langle j \rangle [1 - F_D \cos(2\pi \nu_{\mathrm{mod}} t)] && (27)\\
F_D &= D \;\equiv\; \frac{j_0/2}{\langle j \rangle} && (28)
\end{aligned}
$$

where for the PIC code simulations, $F_D = 1$ for 100% modulated current density was used, so $\langle j \rangle = j_0/2$.

F_R is symmetric about its maximum value located at $\nu_{\text{mod}} = \nu_{\text{nat}}$, and it is easy to define various Full-Width Half-Maximum (FWHM) parameters. The frequencies defining the FWHM values of F_R are,

$$\nu_{\text{mod}_{\text{FWHM}}} = (1 \pm a_6)\nu_{\text{nat}} \tag{29}$$

so the FWHM of the F_R *near*-resonance curve spans

$$\Delta\nu_{\text{FWHM}} = 2a_6\nu_{\text{nat}} \tag{30}$$

and the F_R *near*-resonance curve widens as ν_{nat} increases.

Also, for $\nu_{\text{mod}} = 0$ and $\nu_{\text{mod}}/\nu_{\text{nat}} = 2$,

$$F_R = \frac{a_6^2}{1 + a_6^2} \tag{31}$$

Other relationships involving F_R are

$$\frac{\nu_{\text{mod}}}{\nu_{\text{nat}}} = a_6\left(\frac{1}{F_R} - 1\right)^{1/2} + 1 \quad \text{, or} \tag{32}$$

$$a_6 = \frac{\nu_{\text{mod}}/\nu_{\text{nat}} - 1}{(1/F_R - 1)^{1/2}} \tag{33}$$

Curve-Fitting Rationale

This subsection will discuss the curve-fitting rationale. Most of the information needed to fit the data now exists: the generalized small-spot resonance equations in PIC and UV parameters, the theory values for the constants to be tested, the PIC data, and a near-resonance function F_R for simultaneously curve fitting all of the $\langle P_r \rangle_s$ PIC data. Basically the generalized equations and constants to fit have been presented, along with how the PIC data for the fits were measured. Before presenting and discussing the results of fitting the constants associated with the equations to the PIC data, a more detailed explanation of the curve-fitting rationale and process should prove beneficial. This subsection provides the rationale for using curve fitting to compare theoretical equations with data predicted by those equations.

The rationale for curve fitting the constants in the generalized theory equations depends upon the goal. The reasons people use curve fitting are many. In this case, the fits can be used to summarize the PIC data as a convenient equation, to test the theory-versus-data scaling, to gain clues for helping to improve the theory, and for prediction and optimization purposes. Since the last half of this paper focuses on comparing the theory versus PIC data, I will concentrate on the testing aspect of curve fitting.

Inherent in this testing process is the advantage of having a *useful theoretical equation* that captures the dominant physics of interest in its models, its symbols, and the relationships among those symbols. One then tries to find for that theory equation the *value of any constants* with respect to the PIC data that cause the theory equation to best reproduce the PIC data. Note that *any symbol* in the theory equation, such as a number, coefficient, variable, or exponent, as long as it does not vary with the PIC data, can be treated as a constant that one can attempt to fit with the available PIC data. And when a theoretical equation is used, the equation and the form of the equation showing the mathematical relationships among the symbols remains fixed.

Consequently, one is not looking for which parameters and for which quadratic, power, or other type of function best interpolates the data. Instead of being just an arbitrary equation, the fitting equation is chosen to express inherent theoretical parameters, the relationships among those parameters, and the variation of some chosen dependent variable with respect to all of the other parameters. In essence, the theoretical fitting equation is chosen to represent the *theoretical scaling laws* that capture the important relationships with respect to, and among, the various theory parameters. If the theory and data are good in a given region of parameter space that is of interest, then *the fitted coefficients can be meaningful, and the equation can be extrapolated* to an extent that depends on how well the theory and data match. If a good fit is found, the fit can be used to improve the theory, to make better predictions, and to find the optimum feasible parameters. By curve fitting the equation, a universally applicable way of testing the data versus the theory is utilized. As a result, any constant coefficients, exponents, or parameters in the theory equation can be fitted, thereby providing scaling laws based on data that can be tested versus theory predictions. The testing can reasonably be performed as long as both the theory and data are adequate for the testing purposes.

Of course, one could simply plot the data points versus the theoretical predictions to test the theory. However, a qualitative graph is less quantitative than a curve-fit analysis. Also, qualitative graphical testing is hampered for complex multi-dimensional equations and data since it is hard to graph more than three or four dimensions simultaneously. A 2-D graph could be used that plots a subset of the multi-dimensional parameter space, but this decoupling of parameters can make it more difficult to use available data or obtain enough good data to adequately test and interpret the combined effect of many interacting parameters. To be done unambiguously, testing the theory versus data is best accomplished by using all of the available data to simultaneously analyze as many fitting constants as possible in the theory equations. Further, this testing is *best* accomplished when good data are available that allow fitting *all* of the possible theory equation fitting constants simultaneously. So when using a large number of parameters, testing theory predictions versus data is generally better accomplished through curve-fit analysis than by simply plotting a theoretical curve on top of either computational or laboratory experimental data, though the concepts are similar. Consequently, numerical curve fitting is used because curve fitting a theoretical equation is not only more quantitative than using a graphical technique, but for a large number of parameters, curve fitting is preferable as it allows all of the degrees of freedom to be analyzed simultaneously to separate out the interacting, individual behaviors of the constants and parameters.

On a more practical basis, curve fitting all of the possible constants and parameters simultaneously means that any data taken from any experiment described by the theory, and which are possible for the experimentalist to obtain, can all be used in the analysis. As such, experiments in small subsets of parameter space, and even totally random experimental data, can all be combined and used in the analysis. Using all of the available data, and being able to build up the database as new data comes in, means that the analysis will provide better results than if individual, smaller sets of the data only were able to be analyzed. Better results are possible because simultaneously analyzing all of the parameters, allowing any and all data to be used, produces a synergistic effect. Provided the theory and data are useful, an improved global fit can occur, with better and potentially more parameter scaling estimates being fitted to the theory constants. So, curve fitting is an efficient way to use all available data to test the theory predictions.

Testing of the theory using PIC code simulation data is ideal. The approximations and models used by the theory to derive the theory equations are themselves relatively simple and non-physical. The simple theory models are used because they are approximations of the real world that simplify the physics and make the mathematics tractable. Often a PIC code can closely implement these simple models to test the accuracy of the derivations as well as the effects of implementing the models in an improved and completely self-consistent, nonlinear manner. Also, real world diagnostic measurement problems are avoided or reduced. In fact, PIC code simulations do not suffer from many of the limitations and sources of error that plague a laboratory experiment. Some of these laboratory experimental limitations and sources of error worth mentioning are equipment calibration, limited measurement regimes, environmental influences, and diagnostic tools and probes that themselves interfere with the operation or observation of the experiment. In contrast, PIC code diagnostic probes are essentially 'perfect' and do not interfere with the physics being simulated. Also, the PIC code can easily be set up to look at a wide variety of parameters taken over a large and continuous range of values. This is not to suggest that laboratory experiments are useless or that PIC code simulations are perfect. PIC codes have other problems that need to be addressed when designing a simulation, such as resolving all of the physics adequately, modeling the device or experiment correctly, and maintaining numerical stability to name a few. Still, PIC code simulations complement laboratory experiments and are ideally suited to testing theoretical models because PIC codes can often closely implement theory models, have perfect diagnostics, and can vary all of the parameters continuously over a wider range than the laboratory experiments.

Strictly speaking, in testing the theory versus data using curve fitting, the null hypothesis (H_0) should be tested using a statistical significance test. The curve-fitting process, given knowledge of the errors inherent in the data, can provide an independent goodness of fit criterion against which to judge the fit. But often only the data set, and not its errors, are available. Without independent estimates of the errors inherent in the data to fit, no independent goodness of fit criterion is available, and another significance test will be used. So the assumption or null hypothesis that will be tested is that any apparent deviations or differences arising between the theory equations' predictions versus the PIC code simulation data are only due to random variations from the theory equations' predictions, and are therefore of an insignificant nature. As such, to within theory and PIC differences or 'errors' from each other or from perfect real experiments, *some* of which can be estimated from the fits, the theory equations should predict the PIC data and the PIC data should display the scaling laws inherent in the theory equations. The null hypothesis can be statistically rejected when there is significant and unreasonable disagreement between the theory equations and the PIC data, which could then be considered as different. Accepting or rejecting the null hypothesis does not claim that the decision is always correct, just statistically known for a certain error probability, generally chosen to be 0.05, 0.01, or 0.001, where certain assumptions about the distributions have also been made. As discussed in the previous paragraph, PIC code simulation results should be ideally suited for testing the theory models, as the PIC code can implement most of the theory models perfectly, and then simulate them in a fully self-consistent, nonlinear manner. However, there do exist differences between the theory models and PIC code simulations, and much like in the laboratory, perfect computational or PIC data are hard to obtain. Even with perfect model simulation and perfect PIC data, differences still would be expected to arise due to the PIC code's fully nonlinear, self-consistent computational solutions versus predictions based solely on simple analytic theory.

Consequently, when comparing the theory equations to PIC code simulation data, any careful, well-performed, strict significance testing would likely reject the null hypothesis unless, assuming the theory predictions and PIC code data are inherently different and limited but not basically incorrect, a good way could be found to identify, quantify, and eliminate or include *all* of the total error inherent in both the theory and PIC predictions when doing the significance calculations. Essentially, using only part of the larger total error by estimating the total error from just the sample variance of the fit errors increases the likelihood of rejecting the null hypothesis. So if significance testing of the differences between the theory and fitted equations is performed using only the variance of the fitted equation versus PIC code data as an estimate of the total variance, then significant rejection or borderline acceptance of the null hypothesis could become definite acceptance if the total errors were known and accounted for in the testing. Since it is difficult to identify, let alone quantify, all of the sources of error contributing to the total error, the less strict goal of these comparisons will be to find at best general, though not necessarily statistically significant, agreement among the theory equations and PIC code data with large disagreement expected to occur if the theory equations and PIC data did not generally agree. So for large disagreements, one could conclude that either the theory equations or the PIC data (or both) are probably different. But if the theory equations did generally agree with the PIC results, then while not proving that the equations and PIC data match or are correct, the general agreement would provide satisfying additional confidence in the theory equations.

Agreement between both the simple, limited, and inherently different PIC code results and theory equations' predictions would not only provide confidence that those results are not fundamentally incorrect, but would also provide some measure of confidence that the theoretical and computational results were also representative enough of actual photoelectron-generated radiation to be useful in predicting and optimizing laboratory experiments. Of course inherent differences will exist among the theoretical, computational, and laboratory sample results, as well as each of the sample results having differences from the parent or true results. But the simple, ideal theory and PIC results, by themselves showing agreement, could also indicate that the inherent differences are not so great as to prevent the theory and computational results from being able to also provide useful predictions and scaling laws for laboratory experimental results. However, similar testing of the theory equations versus laboratory data would be needed for more direct confirmation of the usefulness of the theory scaling laws and predictions with respect to laboratory experiments.

In summary, since quantifying the total errors in the theory and PIC and postprocessed data is not easy, significance testing of the null hypothesis is difficult to accomplish accurately. Fortunately, significance testing of the null hypothesis may be less useful to this analysis than other testing considerations. Ideally, *if* the theory did exactly match the PIC data and *if* there were enough well sampled, accurate, and precise PIC data, then regardless of the cost function used the fitted constants should consistently match the theory constants, and the fit errors should approach zero. But a very simple analytic theory will be tested versus a small set of data that at times were coarsely sampled and that were generated by fully self-consistent, nonlinear PIC simulations that contain various inconsistencies with the theory models as well as potentially significant numerical solution errors. Consequently, instead of looking for an exact match between the theory and PIC data set, it is more productive to see how close the two do agree and whether any differences are acceptable. Realistically, both the fitted theory equation predictions and the fitted theory equation constants should reasonably approximate the derived theory equation values so that the fit errors should be reasonably small.

Therefore, for the theory versus data set agreement to be considered good, the fitted equations should approximately reproduce the theory equations, any large variations of the fitted constants from the simple theory constants should be reasonable, and the fit errors should be reasonably small.

Curve-Fitting Process

Next I will discuss the curve-fitting process. The fitting process involves topics related to performing the curve fit. It is a somewhat arbitrary split from the curve-fitting rationale discussed above which stressed the concepts involved in testing the PIC data versus theory. Curve-fitting process topics that will be discussed next include good data, the curve-fitting program, and various cost functions.

To be useful to testing and the curve-fitting process, good data meeting several requirements are necessary. These requirements involve taking data for testable theory parameters and predictions. The data points and theory parameters have to correspond to one another to be compared and should ideally capture theoretical or experimental features that are both important to test and optimal for the testing process. As such, important considerations include: how many parameters are involved, which parameters can or need to be tested, how many data points are needed, where are the data to be sampled, and what are the accuracy and precision of the measured data? These requirements are inherent in the theory, since in order to test whether the predicted behavior of the theory is correct, the theory itself tells us what parameters must be studied, where the parameters and predictions can best be tested, what is the range of parameter space that must be explored, and what trade-offs are involved. Trade-offs exist among the number of parameters being tested, the nature and extent of the theoretical scaling laws being tested, the number of data points needed, where the data are measured in parameter space, and the error inherent in the measurements. So good data that overlap the theoretical scaling being tested well enough to extract useful information from the fitting process are essential in simultaneously solving for the theoretical scaling constants and in testing the theoretical scaling laws.

Several items should be mentioned about the curve-fitting program that I wrote and used to compare the theory and PIC data. The program used is basically a Monte-Carlo code *with added powerful heuristics* that make it much more efficient than a pure Monte-Carlo code. A Monte-Carlo technique was used because it can *directly* handle complicated nonlinear equations, diverse cost functions, and an 'auto-no-slack' feature. While not needed for this analysis, the fitting code can even handle discontinuous functions and algorithms. So the fitting program operates on nonlinear equations or systems of equations directly and does not require linearized equations and the necessary adjustments to the cost function that linearized equations require. However, linear elements in the nonlinear equation can be calculated analytically, provided the standard least squares cost function is used. Otherwise, any cost function can be used, and several are.

The program can use as much information as is available to help constrain the solution. Constraints or information can be in the form of the data set itself, ranges and limits on the constants being fitted, the parameter set selected such as using the PIC parameter set equations which have fewer unknowns, and estimated or best guess values for constants. The program is interesting because it can use various constraints, any set of functions or algorithms, as well as any cost function, while at the same time solving for underdetermined, determined, and overdetermined solutions.

Curve fitting often involves underdetermined and overdetermined solutions to various of the constants when many parameters and complex equations are involved. The program automatically identifies which are the underdetermined or weakly determined constants, what I refer to as slack constants. In the same way, the program also identifies the not-so-slack, or principal, constants required for a good fit without the usual need for forward or backward stepwise regression techniques. This auto-no-slack feature frees the user from constantly adjusting which constants to fit as new data becomes available for various parameters.

Like standard curve fitting, this data-versus-theory comparison program takes the generalized theory equations and fits the constants using the PIC data points. But the process is different from standard curve fitting where only the PIC data would determine the constants being fitted, since the user can also input ideal best guess constant values that the program will try and fit as well. Consequently, the program used to do the fits for this paper can, in essence, also be set to use the PIC data to automatically push the constants away from any user specified theoretical or otherwise determined best guesses, as needed. As such, the program can set any of the constants being fitted to their user inputted best guess values if the data match the theory or if the data for finding those constants are nonexistent or poor. Only when the data do not readily support the ideal constant values of the theory equations do the data push the values of the generalized theory equations' constants being fitted away from their ideal values. Therefore the automatic pushing of the constants away from their user inputted ideal values provides a relatively definite and unambiguous indication of data set versus theory disagreement. So while the program can do, and for this paper usually does do, standard curve fitting, it is more accurate to think of the program as a data-versus-theory comparison tool for answering the question, how different is the data set from the theory?

The fitted results will be presented for several different cost functions. The cost functions typically used in this paper are defined in Table 2 and will be referred to as F1 to F5.

Table 2: Definitions of F1 to F5 Cost Functions

Label	Label's Cost Function and Definition
F1	$\sum \xi_A^2 = \sum$ Squared Values of the Absolute Errors
F2	$\sum \xi_R^2 = \sum$ Squared Values of the Relative Errors
F3	$\sum \xi_A^2 \sum \xi_R^2 =$ F1 multiplied by F2
F4	$\sum \lvert\xi_A\rvert = \sum$ Absolute Values of the Absolute Errors
F5	$\sum \lvert\xi_R\rvert = \sum$ Absolute Values of the Relative Errors

The cost function is what the curve-fitting program minimizes when trying to find the best combination of fitted constants for the generalized equations that reproduce the data. The cost function is calculated in terms of the sum of some type of error, or difference, between the predicted and experimental dependent variable values. For each sum, the predicted dependent variable values are calculated for a given set of experimental data points' independent variable values using the generalized scaling equations and one of the many trial sets of program generated fitting constant values that the curve-fitting program will test to minimize the cost function. Various standard cost functions are typically used due to mathematical and robustness considerations, and I used several others as well. Often these cost functions involve the type of error, either absolute or relative, and whether the value is then squared or taken as the absolute value. The least squares curve fitting most often seen in introductory texts uses the sum of the squared values of the absolute errors as the cost function [Ref. 4, pp. 204-206], while the sum of the absolute values of the absolute errors is considered to be a more robust cost function in that the curve fit is less biased by any outliers in the data set [Ref. 5, pp. 70, 133-137].

By presenting various fits for different cost functions, a more unbiased set of fitted constant values is obtained that can also give a feel for the variability in the fitted constants. However, different cost functions have different characteristics and cannot all be used equally well in all situations. In an ideal situation, where the theory is exact and correct, and the data exactly fit the theory and there are enough data, then any choice of cost function will eventually converge to the correct coefficients. However, even with perfect theory and data the convergence rate can vary with the various cost functions and adversely affect computational efficiency. In a situation that is not ideal, where the data have errors, *a cost function should be chosen that represents and accounts for σ_i, the uncertainties in the y_i data points, as well as the type of error inherent in each of the data points.* Thereby, the effect of the error inherent in each individual data point will be correctly 'weighted' to determine the proper amount of each data point's contribution toward calculating the best values for the fitted constants. But since the total uncertainty in each y_i data point is often unknown, it will be assumed that all of the uncertainties can be described by a single constant such that $\sigma_i \equiv \sigma$. Selecting a constant σ simplified the cost functions used in fitting the constants as well as the calculation of the fit or sample error variance, as a constant σ can be ignored in the minimization process and cancels out in the calculation of the fit variance. However, a constant σ requires the user to select a type of error that is both constant and most representative of all of the total uncertainties or errors of all of the individual data points taken over the entire range of the curve fit. In selecting a cost function that is best suited to fitting the PIC data, I will generally assume that the PIC data points obtained are relatively accurate with respect to each other unless significant measurement and other types of errors external to the PIC code are present or that PIC code uncertainties across PIC simulations, such as for different temporal and spatial resolutions of the varying electron sheaths, are significant. As such, for PIC data points with large and small y_i values, the small values are assumed to be more likely to remain relatively accurate, rather than reflecting random noise or measurement errors of a given size.

For example, if the y_i PIC data point values of the dependent variable are relatively accurate and span several orders of magnitude, a relative error cost function is required, as an absolute error cost function will tend to see the information contained in the smaller data values as noise. This happens to be the case when fitting the nonresonance $\langle P_r \rangle_s$ equation using all of the near-resonance power data. The smallest to largest values of the near-resonance power data span three to four orders of magnitude, so that the smaller values of the radiated power are not well represented by a absolute error fit. Further, the radiated power data include two resonance curves, where the smaller of the two curves is somewhere between two to three orders of magnitude smaller than the larger curve. Because the smaller curve's radiated power information is largely dwarfed by the error of an absolute cost function fit, an absolute error fit largely ignores the resonance frequency and near-resonance curve width information contained in the smaller curve. For the fitting process to fairly weight the equally valid information contained in the smaller radiated powers of the smaller resonance curve, it is necessary to use a relative error cost function. So the F2 cost function, which uses the sum of the squares of the relative errors, would be a better choice than the F1 cost function for fitting and testing the near-resonance radiated power data.

Based on the previous arguments, this paper will generally consider F2 to be the overall best PIC data curve-fitting cost function. However, there are exceptions. For instance, the data sampling of the oscillating d_{max} and $E_{Z\ \text{max}}$ observables was very coarse and resulted in fitted constants and fit errors that reflect the coarse sampling of those oscillating values. Further, the range of the d_{max} and $E_{Z\ \text{max}}$ values is not large. So while the PIC simulation results are themselves often relatively accurate, it still can be difficult to predict which cost function should be considered the *a priori* best for d_{max} and $E_{Z\ \text{max}}$ due to sampling technique or measurement errors that could be larger than the inherent relative errors for a given set of data points.

The F3 cost function minimizes both the absolute and relative errors so that one is not minimized at the expense of large errors in the other. Consequently, the F3 fit is good at reproducing the qualitative aspects that one would try to achieve in manually matching a graph to data points, and is generally a more robust cost function than either F1 or F2 alone. In practice, this cost function is useful when the dominant error inherent in the data set being fitted is not well known. Therefore, the F3 cost function has been included in this paper. So, the F3 cost function helps both types of errors to become acceptably minimized, with the minimization mostly effecting the type of error having the most variability in its respective sample standard deviation between the F1 and F2 fits.

But generally the PIC data are relatively accurate if extracted correctly, and can even be accurate in an absolute sense for certain observables and carefully set up PIC simulations. For example, the ν_{nat} data set represents a PIC code simulation observable that is usually fairly accurate, was calculated very precisely with quantified FFT box width errors, and does not span a relatively large range of ν_{nat} dependent variable values. In fact, ν_{nat} data are accurate and precise enough so that the PIC data for ν_{nat} could be good to approximately one gigahertz of error in an absolute sense, and to about FFT resolution in a relative error sense. Also, over half of the radiated power data describe two F_R resonance curves, and the resonance frequency and near-resonance curve width information should be robust across different PIC code simulations. So ν_{nat} and near-resonance width data should be robust across all of the PIC simulations, and should be relatively insensitive to cost function. Consequently, fitting the ν_{nat} data should result in consistent fitted constants that are insensitive to the choice of cost functions, and with errors that rival the FFT resolution.

Finally, while $\langle P_r \rangle_s$ is calculated relatively accurately within resonance curve runs, and is somewhat relatively accurate across PIC simulations, it may not be particularly accurate in an absolute sense. The postprocessor calculated $\langle P_r \rangle_s$ data may not be particularly accurate in an absolute sense because of such effects as not modeling the actual physics exactly (nor even the theoretical models used in testing), like simulating perpendicular photoemission, and effects of the coarse spatial and temporal resolutions used in simulating and postprocessing the electron sheath to calculate the radiated energy. While needing more study, a few PIC simulations conducted to look at the effect of computational resolution indicate that coarser resolutions in space and time tended to increase the amount of calculated radiation from otherwise similar simulations. This evidence provides one of several possible explanations as to why the fitted radiation equation results discussed in the next subsection seem to be three to four times larger than the theory predictions for a', while still generally reproducing the theoretical scaling laws in the radiation equation's other fitted constants. Fortunately, the other observables that will be discussed in the next subsection were found to be less affected by spatial and temporal resolutions.

Curve-Fitting Results

This subsection will present and discuss the curve-fitting results. The fitting results will be presented in tables. First the small-spot, coefficient and constant fit results will be presented for various equations and will be followed by the small-spot, near-resonance fit results. Coefficient and constant fit results will be presented for $E_{Z\ \mathrm{max}}$, d_{max}, ν_{nat}, and $\langle P_r \rangle_s$ PIC data, whereas near-resonance results will focus on $\langle P_r \rangle_s$ fits. However, fitting the near-resonance $\langle P_r \rangle_s$ equation will sometimes involve simultaneously fitting the ν_{nat} equation for various constants. For fitting self-oscillating and resonance-modulated PIC data, only one or two of the scaling equation constants will be fitted, and auto-no-slack is not used in calculating any of the fitted constants. For the near-resonance data, simultaneous fits involving 2, 3, or 7 of the constants will be presented. Curve fits involving only 1, 2 or 3 of the constants will not use the auto-no-slack feature. Currently, simultaneously fitting 7 of the 9 total constants is the most that the available PIC data can attempt to fit, and even then only enough data exist to fit either a_4 or a_8, but not both. So auto-no-slack is used to a small degree in simultaneously fitting 7 constants but only to fit the three constants a_7, a_8, and a_9. The auto-no-slack feature was primarily designed for analyzing the laboratory's UV parameter data set when it becomes available, with its larger number of parameters and fitting constants in the theory equations and more limited data.

Each table not only shows both the theory equation and the fitted equation constants for several cost functions, but also shows the error statistics associated with each set of theory and fitted equation constant values as well. It is important to note that the error statistics are descriptive and can be used to judge the fitted equations and constants independent of the cost function used to obtain the fitted constants. However, each cost function does minimize its closest associated error statistic, so that minimizing F1 also minimizes the absolute error variance statistic while minimizing F2 also minimizes the relative error variance statistic. The absolute error is calculated as the difference of the fitted equation's value calculated for the dependent variable minus the PIC code data point's value of the dependent variable. The relative error is then calculated as the negative value of the absolute error divided by the fitted equation's value of the dependent variable. Next, the values of these two types of error are averaged over all of the PIC data points for a particular fit to determine the fit's average absolute error and average relative error. If a tabled average absolute error is negative, then the fitted equation is on average located below the data points. Note that the absolute errors have units of volts per meter, meters, hertz, and watts, and are on the order of kilovolts per meter, microns, gigahertz, and nanowatts for the 1-D, small-spot PIC calculations. The sample standard deviations for the absolute (and similarly for the relative) type of errors are correctly calculated using the degrees of freedom available after the curve fit (f_d) as calculated from the number of constants fitted (n_c) and the number of PIC data points used in the fit (N).

The type of errors used in the calculations to perform the curve fits as well as the error statistics presented in the tables will now be expressed as equations and discussed. Let $y_f \equiv y_f(\mathbf{x})$ denote the value of the dependent variable calculated using the fitted equation, where $\mathbf{x} \equiv (x_1, x_2, \ldots, x_{n_v})$ and n_v is the number of independent x_i variables in the fitted equation. Also, for the $i = (1, 2, \ldots, N)$ PIC code $(\mathbf{x}_i, y_i)$ data points, let y_i denote the ith value of the dependent variable. Finally, let ξ_A and ξ_R denote the absolute and relative errors respectively. Then for each PIC code simulation data point,

$$\xi_A \equiv y_f(\mathbf{x}_i) - y_i \quad \text{and} \quad y_i = y_f(\mathbf{x}_i) - \xi_A \tag{34}$$

$$\xi_R \equiv \frac{y_i - y_f(\mathbf{x}_i)}{y_f(\mathbf{x}_i)} \quad \text{and} \quad y_i = [1 + \xi_R] y_f(\mathbf{x}_i) \tag{35}$$

The curve-fitting program minimized the F1 through F5 cost functions to simultaneously determine the optimum values of the various constants in the generalized scaling law equations that were fitted to the PIC code data. In the tables, for each cost function the associated fitted constants and fit errors are listed. Tabled error averages are straight-forward to calculate, and the error sample standard deviations were calculated based on the general definition of the variance of a curve fit being [Ref. 4, p. 187]:

$$s^2 = \frac{\frac{1}{N-n_c} \sum_{i=1}^{N} \left\{ \frac{1}{\sigma_i^2} \left[y_i - y_f(\mathbf{x}_i) \right]^2 \right\}}{\frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma_i^2}} \tag{36}$$

Specifically, the fit error statistics were calculated over the PIC code data points as follows:

$$\xi_{\text{Aavg}} = \frac{\sum_{i=1}^{N} \xi_A}{N} \tag{37}$$

$$\xi^2_{\mathrm{Asd}} = \frac{N}{N-n_c}\frac{\sum_{i=1}^{N}\xi^2_A}{N} \tag{38}$$

$$\xi_{\mathrm{Ravg}} = \frac{\sum_{i=1}^{N}\xi_R}{N} \tag{39}$$

$$\xi^2_{\mathrm{Rsd}} = \frac{N}{N-n_c}\frac{\sum_{i=1}^{N}\xi^2_R}{N} \tag{40}$$

where the PIC data's σ_i variances were assumed constant, with $\sigma_i \equiv \sigma$, in deriving the minimization functions and sample error variances for each type of absolute and relative error used. So the average absolute error of the fit is ξ_{Aavg}, the sample standard deviation of the absolute error of the fit is ξ_{Asd}, the average relative error of the fit is ξ_{Ravg}, and the sample standard deviation of the relative error of the fit is ξ_{Rsd}. The above error statistics for each curve fit are listed in the same order in the tables, after the fitted constants, such that for the a' and the a' and c' fit tables the error statistics are abbreviated as "Abs Avg Err", "Abs SD Err", "Rel Avg Error", and "Rel SD Error" respectively. The degrees of freedom left over after a specific curve fit, and which are available to use in calculating the sample variance of the fit errors, are calculated as $f_d = N - n_c$, where again, N represents the number of PIC code data points used in a specific curve fit and n_c represents the corresponding number of constants that were fitted.

Since all of the curve-fitting result tables show not only the fitted constants but the corresponding error statistics for each fit as well, these tables can be used to make statistical predictions of the probability of seeing various y_f dependent variable values for a given set of independent parameter $\mathbf{x}$ values. Consequently, the fitted equation's *sample variance* error statistics can be used to create Monte-Carlo scatter graph predictions, or simply to determine standard deviation or confidence interval curves, about the fitted equation's predictions which represent the *sample average* estimate of the true average. For example, using the F1 fit and a student's t-distribution with $t_{\alpha,f_d} \equiv t(\alpha, f_d)$ set to some value determined by selecting a desired central, or two-sided, error probability (α) and by using the degrees of freedom (f_d) of the F1 fit, then the statistically predicted value of the dependent variable y_p can be expressed as

$$y_p = y_f \pm \frac{t_{\alpha,f_d}}{\sqrt{N}}\xi_{\mathrm{Asd}} \tag{41}$$

So the parameter t_{α,f_d} in the above equation allows one to specify a desired confidence interval, above and below the fitted mean value of the dependent variable (y_f), to use in creating the predicted confidence interval curves (y_p). Therefore, for all of the tables, it is now easy to use y_f and the error statistics to generate the $y_p = y_f$ predicted average curve, as well as any predicted confidence interval desired about y_f.

Tables 3-6 and Tables 8-11 show fit results for various of the generalized theory equation constants. Tables 3-6 only fit the a' coefficients for the various curve-fit equations, while Tables 8-11 fit both the a' and c' constants. I will refer to Tables 3-6 fits as coefficient fits and to Tables 8-11 fits as constant or hybrid fits. Tables 3-6 and Tables 8-11 are arranged to show $E_{Z\ \mathrm{max}}$, d_{max}, ν_{nat}, and $\langle P_r \rangle_s$ fit results respectively. All the fit results in Tables 3-6 and Tables 8-11 are 'standard' curve fits which only minimize the F1-F5 cost functions and do not use the auto-no-slack feature to adjust the cost functions. Also, all tables show the values of the fitted constants and the error statistics they generate when used in the captioned theory equations to generate data to compare with the PIC code data. Lastly, all tables show the errors generated when just the theory constants are used. These pure theory errors act as yardsticks for comparing the pure theory with th? data and for comparing the theory equations with the fitted equations and the effect of different cost functions and constants being fitted.

All of the PIC code simulations were performed at only two levels of emitted electron energies, and all of the tables show the results of curve fitting *all* of the resulting, usable 0.2 and 1.3305 eV small-spot data points to calculate the fitted constants. However, Tables 6 and 11 had no 0.2 eV data to fit. Also, Tables 3-5, 8-10, and 15 only had a single 0.2 eV data point to use in the fits. Unfortunately, this *single* 0.2 eV data point and the few other 1.3305 eV data points only provide the necessary *minimum* of two levels of β needed to fit b', so while b' could have been fitted, it was not. Basically, it was decided that not enough points at 0.2 eV (let alone at other levels of eV_b) were simulated with the PIC code to *accurately* fit b' (except for Table 14, and indirectly in Tables 12 and 13) for comparison with the theory. So while all of the applicable 0.2 and 1.3305 eV small-spot data were used in fitting the a' and c' constants, due to the small number of points to fit, the errors involved, and having only one extra β point at a different β level, the b' constant was not fitted. Consequently, simultaneously fitting a', b', and c' to all of the applicable data would be too unreliable. Tables 6 and

11 show the results from fitting 5 data points, and Tables 3-5, 8-10, and 15 each used 6 data points to calculate the curve-fit results (and Tables 12-14 used 41 data points), where the sixth point is the single 0.2 eV data point that was available for those fits. Finally, Tables 6 and 11 used resonance radiated power data in calculating the fitted constants, while Tables 3-5, 8-10, and 15 used self-oscillating data (and Tables 12-14 show fit results for all usable radiated power data).

Tables 3-6 are ideal for comparing the theory predictions for small-spot, steady-state, resonance conditions with PIC data from simulations that implemented many of the theory models. By directly comparing the theory value of the a' constant with the various fitted values, and by noting the respective errors, one can get a good feeling for how well the simple analytic theory equations match the PIC code results. Other items to look at for all of the tables showing curve-fit results are the consistency of the fitted constant values and resulting errors plus the nature of the data used in the fitting process, such as the data's approximate accuracy, precision, and number of points. By understanding the data, the theory, and the size of the fit errors, as well as the closeness and consistency of the errors and fitted constant values, and how they compare to the theory, conclusions can be drawn as to the general goodness of the fit. Of course, the comparison of the theory equations with the PIC data should also take into account the fact that the simple theory was developed to make rough predictions and to develop the general scaling laws for use with testing, understanding, and optimizing observables of interest, and for using the PIC and laboratory data to improve the simple theory results. In a previous subsection, the problems with significance testing were discussed. Briefly restated, as a consequence of the differences and errors inherent in the theory and the PIC code simulations, the theory does not exactly match the PIC code simulations. Therefore, the best that can be expected of the theory-versus-PIC comparisons is that the fitted constants should be close to the theory values, any large deviations of the fitted constants from the theory values should be reasonable, and that the size of the fit error statistics should be reasonable as well.

Several numbers based on Tables 3-6 will now be presented for $E_{Z\,\mathrm{max}}$, d_{max}, ν_{nat}, and $\langle P_r \rangle_s$ respectively. From the relative average error statistic ξ_{Ravg} obtained for the data versus theory, the average ratio between the PIC code data and the y_t theory curve is given by $\langle y_i/y_t(\mathbf{x}_i)\rangle = \xi_{\mathrm{Ravg}} + 1$, or for the various curves by 2.04, 0.939, 0.913, and 3.84 respectively. As

Table 3: a' Coefficient Fits and Errors of $E_{Z\ \mathrm{max}} = a'\beta^{b'}\langle j\rangle^{c'}$

Cost Func	a' Fitted	b'	c'	Abs Avg Err, kV/m	Abs SD Err, kV/m	Rel Avg Error	Rel SD Error
Theory	13875	0.5	0.5	-49.326	54.415	1.0352	1.0449
F1	27767	0.5	0.5	-1.0921	10.054	.016950	.079975
F2	28376	0.5	0.5	1.0201	10.378	-.0048523	.076307
F3	28131	0.5	0.5	.17144	10.171	.0037942	.076895
F4	27241	0.5	0.5	-2.9203	10.298	.036608	.088857
F5	29259	0.5	0.5	4.0856	11.869	-.034883	.083128

Table 4: a' Coefficient Fits and Errors of $d_{\mathrm{max}} = a'\beta^{b'}\langle j\rangle^{c'}$

Cost Func	a' Fitted	b'	c'	Abs Avg Err, μm	Abs SD Err, μm	Rel Avg Error	Rel SD Error
Theory	18.415	1.5	-0.5	3.8364	8.4950	-.060880	.21411
F1	14.137	1.5	-0.5	-2.1489	5.8943	.22332	.38164
F2	18.120	1.5	-0.5	3.4239	8.9273	-.045599	.23392
F3	16.285	1.5	-0.5	.85695	6.9153	.061928	.26317
F4	13.373	1.5	-0.5	-3.2174	6.0328	.29319	.44613
F5	18.554	1.5	-0.5	4.0307	9.4879	-.067911	.23526

Table 5: a' Coefficient Fits and Errors of $\nu_{\mathrm{nat}} = a'\beta^{b'}\langle j\rangle^{c'}$

Cost Func	$a'/10^6$ Fitted	b'	c'	Abs Avg Err, GHz	Abs SD Err, GHz	Rel Avg Error	Rel SD Error
Theory	4.0700	-0.5	0.5	.66615	.75097	-.087263	.090287
F1	3.6700	-0.5	0.5	-.039241	.17807	.012229	.031177
F2	3.7172	-0.5	0.5	.044121	.20178	-.00064419	.027803
F3	3.6923	-0.5	0.5	.00012968	.18362	.0061083	.028771
F4	3.7062	-0.5	0.5	.024612	.19234	.0023391	.027995
F5	3.7321	-0.5	0.5	.070414	.21748	-.0046370	.028146

Table 6: a' Coefficient Fits and Errors of $\langle P_r \rangle_s = a' F_D^{d'} A^{e'} \beta^{b'} \langle j \rangle^{c'}$

Cost Func	a' Fitted	b'	c'	d'	e'	Abs Avg Err, nW	Abs SD Err, nW	Rel Avg Error	Rel SD Error
Theory	24.657	2	2	2	2	-107.48	129.43	2.8446	2.9114
F1	74.722	2	2	2	2	-12.693	24.139	.26866	.37751
F2	97.261	2	2	2	2	29.980	68.620	-.025334	.17795
F3	78.693	2	2	2	2	-5.1748	26.660	.20464	.31543
F4	70.539	2	2	2	2	-20.613	26.922	.34389	.45443
F5	100.63	2	2	2	2	36.366	77.692	-.058002	.18176

such, the values of the the PIC code data are on average about 2.04 times larger than the values of those same dependent variables when generated from the theoretical curve for $E_{Z\ \mathrm{max}}$. So while the average PIC data values as compared to the theory equations are about right for d_{max} and ν_{nat}, the values of the PIC data are on average larger for the other observables. When these same ratio factors are calculated for all of the fitted equations using the various cost functions in each table, where $\langle y_i/y_f(\mathbf{x}_i)\rangle = \xi_{\mathrm{Ravg}} + 1$, and are multiplied with the fitted a'_f to adjust the fitted curve to compensate for the average relative error, the adjusted a'_f is largely independent of the different cost functions and results in adjusted a'_f values of 2.82×10^4, 17.3, 3.71×10^6, and 94.8 respectively. Unfortunately these new average values, if used to recalculate the error statistics, would generally cause the ξ_{sd} error statistics to increase even though the ξ_{avg} error statistics would decrease, since to try and further decrease both the ξ_{avg} and ξ_{sd} error statistics would require an ad hoc extra degree of freedom in the theory equations to represent the average offset error, such that $y = a'\beta^{b'}\langle j\rangle^{c'} + k'_{\mathrm{offset}}$ for example. Since the PIC code nonlinear solution of the problem is self-consistent, and since *the average differences of the PIC data versus the theory do not always show consistent variations from the theory*, the average differences tend to highlight the approximate nature of the simple theory. *Still, the simple scaling laws work better than might have been expected, being well within an order of magnitude in predicting the PIC simulation results.*

The χ^2 chi-square *independent* goodness-of-fit criterion is [Ref. 4, p. 202]:

$$\chi^2 = \sum_{i=1}^{N} \left\{ \frac{1}{\sigma_i^2} \left[y_i - y_f(\mathbf{x}_i) \right]^2 \right\} \tag{42}$$

and requires knowing σ_i for the data points, where again, $\sigma_i \equiv \sigma$ was set equal to a constant as used for the curve fits in this paper. Note that χ^2 is the least-squares cost function that determines the optimum values of the fit constants when it is minimized with respect to each of the fit constants simultaneously. Since σ is generally unknown for the PIC data, and therefore generally unavailable for a χ^2 test, a simple significance test using the student's t-distribution for small f_d degrees of freedom can be used instead to get a rough ideal of the merit of the null hypothesis (H_0). Let t_c be the calculated test statistic. Then using the F1 curve-fit results for y_f, along with the derived theory equation and theory constants for y_t,

$$t_c = \frac{|\langle y_f - y_t \rangle|\sqrt{N}}{s} = \frac{|\bar{y}_f - \bar{y}_t|\sqrt{N}}{s} = \frac{|\xi_{\mathbf{Aavg\ F1}} - \xi_{\mathbf{Aavg\ t}}|\sqrt{N}}{\xi_{\mathbf{Asd\ F1}}} \tag{43}$$

where the fitted equation's dependent variable $y_f = f(\mathbf{x})$ represents the sample mean value for the dependent variable estimated from curve fitting the values of the $(\mathbf{x}_i, y_i)$ N PIC data points. The population mean value for the dependent variable is, for the purposes of this test, assumed to be the theory equation's $y_t = f(\mathbf{x})$. Finally, $\bar{y}$ represents the respective fit and theory equation dependent variable values averaged over the N PIC data points, calculated as

$$\bar{y} = \frac{\sum_{i=1}^{N} y(\mathbf{x}_i)}{N} \tag{44}$$

Table 7 summarizes the significance test results obtained from using the tabled theory-versus-fitted equations' F1 error statistics for the a' only curve fits. If $t_c > t_{\alpha,f_d}$, for a given t_{α,f_d} as determined by both the degrees of freedom of the fit f_d and by the selected error probability value of α, then there are significant differences between the theoretical mean values of y_t and the estimated mean values of y_f, and therefore that there are significant differences between the theory and the PIC code simulation results. Consequently, if $t_c > t_{\alpha,f_d}$, then the null hypothesis must be rejected. So based on this significance test of the a', F1 fits, the null hypothesis is rejected for the $E_{Z\ \mathbf{max}}$, $\nu_{\mathbf{nat}}$, and resonant-driven $\langle P_r \rangle_s$ comparisons, and is accepted for the $d_{\mathbf{max}}$ and self-oscillating $\langle P_r \rangle_s$ comparisons between the theory predictions and computational PIC code simulation data. However, the tabled significance tests do not include all of the total differences between or total errors in the theory predicted and PIC code simulation calculated (and experimental and true) results, so any careful significance testing based only upon the fitted error statistics is expected to show significant differences if enough good data points are sampled and fitted. So even large, rejected t_c values could be acceptable since the total error was not used, and despite the differences, the theory and computational PIC data are both based on simple, fundamental physics and still appear to be useful in predicting and optimizing the various dependent parameter values. In summary, this look at the curve-fit results in Tables 3-6 (and 15) finds the expected general, but not usually statistically significant, agreement among the theory predictions and PIC data.

Table 7: Significance Testing Results of F1, a' Fits Versus Theory

Equation Tested	Table Used	N	n_c	f_d	t_c	t_{α,f_d} $\alpha = .05$	t_{α,f_d} $\alpha = .01$	t_{α,f_d} $\alpha = .001$	H_0
$E_{Z\ \max}$	3	6	1	5	11.751	2.571	4.032	6.869	R
$d_{\max}$	4	6	1	5	2.4873	2.571	4.032	6.869	A
ν_{nat}	5	6	1	5	9.7032	2.571	4.032	6.869	R
$\langle P_r \rangle_s$	6	5	1	4	8.7804	2.776	4.604	8.610	R
$\langle P_r \rangle_s$	15	6	1	5	0.3017	2.571	4.032	6.869	A

The next few paragraphs will examine the fits presented in Tables 3-6 and Tables 8-11 with respect to each fitted observable, in more detail. Various explanations will be put forth as to how some of the differences from the theory could have occurred. The general types of explanations that make the theory-versus-PIC disagreements seem more reasonable are listed below. Sometimes the way the PIC code performed the simulations, or how the postprocessor worked, can be invoked to help explain various results, such as for $E_{Z\ \max}$ and $\langle P_r \rangle_s$. Other times, the behavior of the observable being sampled, the coarse sampling, and limited number of data points can be cited, as for $E_{Z\ \max}$ and $d_{\max}$. Even data that are fitted well, such as for ν_{nat}, cause the number of error standard deviations to be large by the very fact that the fits are generally consistent and good, having small errors.

This brings up the question of the overall accuracy and precision of the PIC data obtained, how well the PIC simulations and resulting PIC data match the theory models the data are being compared to, and whether the PIC data set for the models used is itself accurate in a relative, absolute, or both relative and absolute sense. In general it should be stressed that while

Table 8: a' and c' Constant Fits and Errors of $E_{Z\ \mathrm{max}} = a'\beta^{b'}\langle j\rangle^{c'}$

Cost Func	a' Fitted	b'	c' Fitted	Abs Avg Err, kV/m	Abs SD Err, kV/m	Rel Avg Error	Rel SD Error
Theory	13875	0.5	0.5	-49.326	54.415	1.0352	1.0449
F1	47972	0.5	0.44013	.51470	10.033	-.024340	.10500
F2	29960	0.5	0.49354	.72722	11.233	-.0048192	.085023
F3	34516	0.5	0.47652	-.18495	10.511	-.0029576	.087053
F4	42960	0.5	0.44845	-3.0216	11.006	.015370	.10165
F5	42960	0.5	0.44845	-3.0216	11.006	.015370	.10165

Table 9: a' and c' Constant Fits and Errors of $d_{\mathrm{max}} = a'\beta^{b'}\langle j\rangle^{c'}$

Cost Func	a' Fitted	b'	c' Fitted	Abs Avg Err, μm	Abs SD Err, μm	Rel Avg Error	Rel SD Error
Theory	18.415	1.5	-0.5	3.8364	8.4950	-.060880	.21411
F1	2.4109	1.5	-.28181	-1.1236	4.8363	.16019	.50874
F2	18.622	1.5	-.50323	3.4400	10.051	-.045588	.26150
F3	6.8397	1.5	-.39852	.27814	5.9348	.077375	.33978
F4	1.6004	1.5	-.23703	-1.7793	5.0209	.20954	.60546
F5	22.204	1.5	-.52032	4.3418	11.284	-.074925	.26491

Table 10: a' and c' Constant Fits and Errors of $\nu_{\mathrm{nat}} = a'\beta^{b'}\langle j\rangle^{c'}$

Cost Func	$a'/10^6$ Fitted	b'	c' Fitted	Abs Avg Err, GHz	Abs SD Err, GHz	Rel Avg Error	Rel SD Error
Theory	4.0700	-0.5	0.5	.66615	.75097	-.087263	.090287
F1	4.6716	-0.5	.47327	-.0060936	.12645	.0014293	.025464
F2	4.3934	-0.5	.48059	.015210	.13873	-.00039380	.024304
F3	4.5629	-0.5	.47600	-.0024320	.12774	.0013984	.024825
F4	4.8246	-0.5	.46968	-.0021617	.12817	.00014079	.026600
F5	4.8246	-0.5	.46968	-.0021617	.12817	.00014079	.026600

Table 11: a' and c' Constant Fits and Errors of $\langle P_r\rangle_s = a'F_D^{d'}A^{e'}\beta^{b'}\langle j\rangle^{c'}$

Cost	a'	b'	c'	d'	e'	Abs Avg	Abs SD	Rel Avg	Rel SD
Func	Fitted		Fitted			Err, nW	Err, nW	Error	Error
Theory	24.657	2	2	2	2	-107.48	129.43	2.8446	2.9114
F1	4383.3	2	1.5726	2	2	.59156	4.0681	-.022356	.068135
F2	2113.9	2	1.6516	2	2	1.3256	8.6550	-.0010763	.042353
F3	3231.7	2	1.6046	2	2	-.52564	4.6785	-.0049996	.050839
F4	4835.9	2	1.5618	2	2	.37308	4.2345	-.024276	.073370
F5	2246.3	2	1.6471	2	2	4.3117	11.861	-.021425	.049914

the PIC data and sampling are not necessarily perfect, the simple theory itself may be the limiting factor in trying to make too close of a comparison with the generally better, self-consistent PIC code modeling and solutions. Therefore, this paragraph ends with a warning not to expect to find *too* close of an agreement or consistency between the theory and PIC results, because both the theory and PIC simulations do have differences and limitations.

$E_{Z\ \mathrm{max}}$ was coarsely sampled data that came directly from the PIC code graphs, and represents six self-oscillating simulations. The fits turn out to be fairly good in the size of the errors, since errors of that size were expected due to the coarse sampling of the variations in the oscillating $E_{Z\ \mathrm{max}}$ peak values. The errors do not clearly indicate that any of the cost functions is necessarily best. In fact, the fitted a' coefficients are fairly consistent, and indicate that the maximum electric field strength predicted by the steady-state theory is about half of what the coarsely sample peak values of the PIC code simulation data suggest.

The increased electric field strength found in the PIC data could be the result of several factors. One explanation is that the increased electric field strength in the simulation is the result of the self-oscillation causing higher peak field strengths to have been sampled. But this is not true, and is shown by one of the PIC simulations, where a damped self-oscillation steady-state E_Z was reached. For this damped self-oscillation simulation, the coarse sampling should have accurately, if not very precisely, measured the average $E_{Z\ \mathrm{max}}$, because there were no significant oscillations affecting the last sampled graph that was dumped in time, from which these data were taken. Calculating the value of a' for this essentially non-oscillating E_Z data *still* results in a single-point solution $a' = 29259$, that closely matches the 6 point curve fit and is also about twice as large as predicted. Since the single point non-oscillating solution for a' represents the average $E_{Z\ \mathrm{max}}$ and agrees with the 6 point curve-fitted solutions, this indicates that the coarse sampling of $E_{Z\ \mathrm{max}}$ turns out to have been somewhat randomly distributed about the average values of the oscillating $E_{Z\ \mathrm{max}}$. Consequently, it is more likely that the higher than expected electric field strengths are due to the inherent differences between the simple theory and PIC code simulation results. Two examples of such differences in the PIC code simulations versus the theory results are: 1) the fully self-consistent, nonlinear PIC code solutions, and 2) the PIC code not returning the charge of the initially emitted electrons that escaped the simulation boundary at $z = 100$ μm, to the emitting conductor. Note that Figs. 4-6 indicate that roughly 2 kV/m (0.2 eV) and 14 kV/m (1.3305 eV) E_Z field strength increases occurred due to the escaped charge. When treated as systematic errors and subtracted from the $E_{Z\ \mathrm{max}}$ data given in Table 16, the new a' fit using the F2 cost function calculates that $a' = 24637$, with $\xi_{\mathrm{Ravg}} = -.0057977$ and $\xi_{\mathrm{Rsd}} = .083410$. As such, compensating for the escaped charge appears to only result in a 13.2% decrease from the previous F2 fitted a'. However, future simulations would help to confirm the causes and the extent to which each cause contributed to the difference between the theory and PIC data.

The poorly sampled d_{max} PIC data come directly from the PIC code, being measured from graphs that MAGIC created. The dominant type of error inherent in the data is difficult to identify for determining which of the cost functions is necessarily best. However, due to the nature of the data set with its limited number of data points and relatively large errors, all of the cost functions result in fitted constants that are generally close to, though sometimes less than, the predicted theory value, and that have fit errors reflecting the expected sampling error. Further, because the sampling error is somewhat large, along with the fact that only six data points exist, allowing the c' constant to also be fitted in Table 9 does not give very consistent results. In essence, either more accurate data or more data of the same accuracy would be needed to predict a more consistent set of fitted constants in Table 9. Still, the constants in Table 9 do show the right trends. Finally, as noted in discussing $E_{Z\ \mathrm{max}}$ above, the electric field strengths simulated by the PIC code do indeed seem to be about twice as large, on average, than predicted by the theory. This should have the effect of decreasing the *average*

d_{max} by half, which is only partially supported by the values of the fits to the oscillating d_{max}. As was done for the average electric fields, the damped Sun workstation PIC simulation data for d_{max} can be used to calculate a steady-state, one-point solution for $a' = 11.355$. This value of a' corresponding to the steady-state damped solution is about half of what was predicted and is nearly consistent with the doubled electric fields.

But what does the damped solution value for a' indicate? If the damped solution for a' represents the average d_{max}, then since $a' = 11.355$ is lower than *all* of the fitted values for a', this could simply indicate that d_{max} was sampled above the average values for the oscillating d_{max}. But based on all of the momentum graphs, as represented by the momentum graphs in Figs. 2-3, it appears that d_{min} is fairly constant versus steady-state simulation time, and that it is close to the damped solution for a'. Also, the electric fields appear randomly sampled about the average, and the maximum orbits were similarly measured. So based on the above arguments, it is probable that the damped solution for d_{max} of $a' = 11.355$ actually represents the minimum oscillating d_{max}, with d_{max} oscillating about an average a' value somewhere between 11.355 to 18.554 or greater. Unfortunately, the true average orbital distance d_{avg} based on all of the particles in an *oscillating* simulation is hard to determine from these momentum graphs, where potentially a few macro particles, for perhaps short portions of their orbits, and appearing at large d_{max} distances, could bias the results. From the new Cray runs, using the MAGIC graph of the damped simulation in Fig. 2 results in non-oscillating one-point a' solutions for d_{min}, d_{avg}, and d_{max} of approximately 11.39, 11.52, and 11.66 respectively. And using only the parabolic like part of the electron trajectories for the oscillating simulation shown in Fig. 3, the one-point a' solutions for d_{min}, d_{avg}, and d_{max} are approximately 9.966, 10.39, and 10.82 respectively. In conclusion, the average of the *sampled* d_{max} oscillations is represented by the various fits, while the actual average d_{max} could be lower, reflecting increased electric fields, and where the more accurately measured data for *damped a' solutions* for the average $E_{Z\ \text{max}}$ and d_{avg} show nearly consistent deviations from the theory.

ν_{nat} data are probably the most robust and accurate data produced by the PIC code simulations and subsequent radiation postprocessing. For ν_{nat}, values ranged from about 4 to 9 GHz, so there was not even much variation in the range of the dependent variable used in the cost functions. The absolute errors from FFT resolution considerations were the same for this set of 6 PIC simulations, with the ν_{nat} PIC data from each simulation being relatively accurate with respect to other simulations, and could be accurate to within one gigahertz from the true values in an absolute sense. Computationally the FFT box width, which defines part of the total error in the postprocessed ν_{nat} data, was 0.15815 GHz for all of the MAGIC simulations. As a result of the relatively accurate and precise ν_{nat} data having small variations in error, all of the fits listed in Tables 5 and 10 give very similar fitted constant values with small errors that approach the width of the FFT boxes. So from Table 5 results based on fitting 6 PIC data points, the estimated value for the a' constant is close to the theory value, the fits are all very consistent, and the fit errors are very good. It would appear that the simple theory equation for ν_{nat} works surprising well despite the rough approximations used to convert the orbit of a 'virtually' thin, rigid, single-pulse charge sheet into a steady-state, pure sinusoidal current source having a predicted orbital frequency. However, the curve-fit deviations from the simple theory are not even fully self-consistent among all of the different generalized theory equation fit results (such as the electric field strengths increasing and electron orbital distances decreasing, but the natural frequency not increasing), so any very close agreements or consistent deviations among some of the theory and fitted equations could be just coincidental.

In comparing the $\langle P_r \rangle_s$ fitted constants with theory, the $\langle P_r \rangle_s$ fitted constants reflect the more complex nature of the $\langle P_r \rangle_s$ data. As seen in Table 6, the fitted coefficient appears to be three to four times larger than the theory constant. Also, the theory curve lies below the data points by an average of about 107 nW. In looking at the errors, the fitted constant values cause the error statistics to decrease, especially for the relative errors. $\langle P_r \rangle_s$ data are not as robust and accurate as the ν_{nat} data, as the coarseness of the spatial and temporal resolutions used in the PIC code and the postprocessor affect the results. And preliminary test runs designed to look into this problem indicate that coarser simulation grid resolutions increase the calculated power. Also, the theory is not only simple, but may actually represent the lower values of the self-oscillation radiated power (see Table 15), rather than the larger values of the externally imposed $\nu_{\text{mod}} = \nu_{\text{nat}}$ driven resonance effects and resulting powers. In addition, Table 11 indicates that the value of the c' constant is also too large, and that the data have a slightly less than quadratic scaling with respect to the current density of the emitted beam (which is also the scaling for the power of the light source when using laboratory parameters). Still, $\langle P_r \rangle_s$ behaves approximately as expected and the simple, limited nature of the theory, as well as simulation modeling, resolution, and goodness-of-data issues, probably account for most of the differences observed between the simple analytic theory predictions and the PIC code simulations.

You may have noticed that the F4 and F5 cost function fit results are identical in Table 8, as well as in Table 10. They are identical because the F4 and F5 cost functions use the sum of the absolute values of the errors (and not the sum of the squared values of the errors) to determine the fitted constants. A result of using the sum of the absolute values of the errors as the cost function is that it creates a fitted equation that exactly intersects the number of data points necessary for a minimum solution, while giving the best fit error from the remaining points that were not intersected by the fit [Ref. 5, pp. 135-137]. With only six points and two constants to fit, two points were fitted exactly and it was not unlikely that sometimes the F4 and F5 cost functions would both select the same two points for minimizing the sum of the absolute values of both the absolute and relative errors. As stated previously, the error statistics are descriptive and can be interpreted separately from the cost functions. However, each cost function has caused its related fitted error sample standard deviation to be minimized, even for the cases where the cost function results in discretely intersecting a subset of the data points.

Also for the same reasons given in the previous paragraph, note that the single point $E_{Z\ \mathrm{max}}$ solution for a' using the damped self-oscillation data point is the same a' fitted solution seen in Table 3 where all six data points are fitted using the F5 cost function, and where only one data point is exactly intersected. Not unexpectedly for randomly sampled data about a mean, especially where a few points are involved, the point intersected as the best solution to minimize the total error from all the data points turns out to be, for the F5 cost function in this case, the steady-state damped one-point solution that represents the average value of the sampled $E_{Z\ \mathrm{max}}$. A match between the one-point direct solution and multi-point curve-fit solution for a' did not occur with the d_{max} data because the *minimum* d_{max} solution for a' seems to have been calculated from the one-point damped simulation, and this solution did not minimize the total fitted error for either the F4 or F5 cost functions, since the sampled data were always greater than or equal to the one-point solution. Consequently, the F4 and F5 fits represent the larger, average values of $d_{\max}$ that were sampled.

In summary, Tables 3-6 and Tables 8-11 provide useful curve-fit results for comparing the theory and fitted equations' constant values and fit errors. The values in Tables 3-6 are especially useful for direct comparisons of how well the theory equations, with their theory constants for parameter scaling, compare to theory values. And by analyzing the results of Tables 8-11 along with the results shown in Tables 3-6, greater insights into the nature of the data set as it compares with the theory were gained. The results are generally quite good, showing that the theory predictions agree in sign and general magnitude with the PIC code simulation data. In one sense, the $E_{Z\ \max}$ fit showed the worst agreement of the PIC data with theory equation predictions, having the worst $t_c/t_{\alpha,f_d}$ significance test ratio for any given α error probability. In another sense, the worst comparison comes from the more complex $\langle P_r \rangle_s$ equation, where the PIC fitted curve turns out to be three to four times larger on average than the theory curve values. That difference is not bad, and considering the simple nature of the theory, along with results of preliminary testing of resolution effects, these differences seem acceptable. Later, Table 15 will take another look at explaining this theory versus PIC data radiated power difference. In conclusion, Tables 3-6 and Tables 8-11 show that the theory predictions and PIC simulation data do not usually agree in a significant way (based on a simple significance test that used only the fitted equation estimates of the total error), but that the agreement is generally good and within an order of magnitude on average, which is quite good when all the limitations, errors, and model differences of the simple theory and PIC simulations are considered.

Tables 12-14 use most of the PIC simulation radiated power data points to obtain additional fitting results. As previously discussed, to do so required an ad hoc F_R resonance function. Consequently, only data points in the first resonance maximum could be used ($0 \leq \nu_{\text{mod}} \leq 2.3\nu_{\text{nat}}$), allowing 41 of the 48 total PIC data points to be used. As a result, the ad hoc Lorentzian function used should provide for a better ν_{nat} estimate of constant a_7, as well as providing an approximate estimate of constant a_6. Also, fitting all 41 data points allows a_4 to be reliably determined. Of course, one can then check how other constants take on fitted values when using the ad hoc F_R function and 41 data points by comparing previously fitted constants with those same constants fitted to 41 data points. The fit errors are calculated as before. Note that I only use the F1, F2, and F3 cost functions in these tables. Again, the F2 cost function estimates for the fitted constants, which more equally weights and uses the information from the smaller 0.2 eV F_R curve, is best for these radiated power fits. Finally, the previous caveats concerning coarse grid resolution and the limited nature of the simple analytic solution, as used in discussing the resonance $\langle P_r \rangle_s$ results from Tables 6 and 11, also apply to the results in Tables 12-14.

Table 12: Small-Spot, Near-Resonance, Constants Fits and Errors of $\langle P_r \rangle_s = a_1 F_D^{a_2} A^{a_3} \beta^{a_4} \langle j \rangle^{a_5} F_R$ where $F_R \equiv a_6^2 / \left[(\nu_{\text{mod}}/\nu_{\text{nat}} - 1)^2 + a_6^2 \right]$, $\nu_{\text{nat}} = a_7 \beta^{a_8} \langle j \rangle^{a_9}$, $F_D = D = 100\%$ and $A = 2.0 \times 10^{-6}$ m^2 were used.

Constants and Errors	Theory Values	F1 Results	F2 Results	F3 Results
a_1 Fitted	24.657	80.751	116.85	83.900
a_2	2.0	2.0	2.0	2.0
a_3	2.0	2.0	2.0	2.0
a_4	2.0	2.0	2.0	2.0
a_5	2.0	2.0	2.0	2.0
a_6 Fitted	None (.62031)	.72546	.62031	.74495
a_7	4.0700×10^6	4.0700×10^6	4.0700×10^6	4.0700×10^6
a_8	-0.5	-0.5	-0.5	-0.5
a_9	0.5	0.5	0.5	0.5
Avg Abs Err	-41.208 nW	-5.1694 nW	12.514 nW	-2.6228 nW
SD Abs Err	68.673 nW	13.803 nW	40.064 nW	14.577 nW
Avg Rel Err	3.4327	.21635	-.064616	.15216
SD Rel Err	3.6250	.40894	.26063	.36535

Three sets of constants were fitted to the 41 PIC data points, resulting in three tables. Table 12 fits the a_1 and a_6 constants, which are generally the minimum number of constants that can be meaningfully fitted. Table 13 fits the a_1, a_6, and a_7 constants for simultaneous estimates of both the $\langle P_r \rangle_s$ and ν_{nat} coefficients. Appropriately, the curve-fit results in Table 13 are similar to the previous curve-fit results for the ν_{nat} and $\langle P_r \rangle_s$ coefficients as presented in Tables 5 and 6 respectively. Table 14 simultaneously fits the a_1, a_4, a_5, a_6, a_7, a_8, and a_9 constants to the 41 near-resonance data points, using auto-no-slack applied to the a_7, a_8, and a_9 constants to act as an additional constraint in helping to determine how a_4 and a_8 should be adjusted for a best fit. From several trial fits, a_4 was found to be more critical and a_8 was found to be more slack. This previous result was automatically enforced by the applied auto-no-slack option.

Table 13: Small-Spot, Near-Resonance, Constants Fits and Errors of $\langle P_r \rangle_s = a_1 F_D^{a_2} A^{a_3} \beta^{a_4} \langle j \rangle^{a_5} F_R$ where $F_R \equiv a_6^2 / \left[(\nu_{\text{mod}}/\nu_{\text{nat}} - 1)^2 + a_6^2 \right]$, $\nu_{\text{nat}} = a_7 \beta^{a_8} \langle j \rangle^{a_9}$, $F_D = D = 100\%$ and $A = 2.0 \times 10^{-6}$ m^2 were used.

Constants and Errors	Theory Values	F1 Results	F2 Results	F3 Results
a_1 Fitted	24.657	80.772	116.82	86.313
a_2	2.0	2.0	2.0	2.0
a_3	2.0	2.0	2.0	2.0
a_4	2.0	2.0	2.0	2.0
a_5	2.0	2.0	2.0	2.0
a_6 Fitted	None (.62031)	.72538	.62074	.72253
a_7 Fitted	4.0700×10^6	4.0723×10^6	4.0667×10^6	4.2928×10^6
a_8	-0.5	-0.5	-0.5	-0.5
a_9	0.5	0.5	0.5	0.5
Avg Abs Err	-41.208 nW	-5.1629 nW	12.535 nW	-2.2706 nW
SD Abs Err	68.673 nW	13.983 nW	40.640 nW	14.742 nW
Avg Rel Err	3.4327	.21590	-.064615	.12492
SD Rel Err	3.6250	.41397	.26404	.36020

The fitting process is being used to compare the simple theory equations with the hopefully better solutions obtained from using MAGIC. Since the constants in Tables 12-14 were fitted with data points spanning several orders of magnitude in the dependent variable, and because the computational errors are mostly relative since the same MAGIC simulation grid was used, the F2 cost function provides the best estimates of the fitted constants. If the PIC and radiation postprocessing calculations are basically correct, even if only in a relative sense among the PIC simulations, then the deviations from the theory constants could largely be due to the fully self-consistent nature of solving the simple theory models using the MAGIC PIC code. For example, in Table 14 the constant a_5 has a fitted value of about 1.77 compared to the theory value of about 2.0, and this difference could be due to the theory models being solved approximately for the analytic theory, while being implemented and solved more directly using the MAGIC PIC code. While the fits and errors do not look exceptionally good, they appear acceptable given various other considerations such as numerical simulation concerns like grid coarseness, and the fact that the theory is itself very simple and does not fully represent nor reproduce the better modeled, nonlinear solution achieved using the MAGIC PIC code. So the data fitted in Tables 12-14 should be relatively correct across PIC simulations and the fitted equations appear to provide generally good agreement and errors when compared with the theory.

Table 14: Small-Spot, Near-Resonance, Constants Fits and Errors of $\langle P_r\rangle_s = a_1 F_D^{a_2} A^{a_3} \beta^{a_4} \langle j\rangle^{a_5} F_R$ where $F_R \equiv a_6^2/\left[(\nu_{\mathrm{mod}}/\nu_{\mathrm{nat}} - 1)^2 + a_6^2\right]$, $\nu_{\mathrm{nat}} = a_7 \beta^{a_8} \langle j\rangle^{a_9}$, $F_D = D = 100\%$ and $A = 2.0 \times 10^{-6}$ m^2 were used.

Constants and Errors	Theory Values	F1 Results	F2 Results	F3 Results
a_1 Fitted	24.657	3.2884×10^5	6.4056×10^3	2.4914×10^5
a_2	2.0	2.0	2.0	2.0
a_3	2.0	2.0	2.0	2.0
a_4 Fitted	2.0	2.6821	2.3307	2.6543
a_5 Fitted	2.0	1.5560	1.7650	1.5677
a_6 Fitted	None (.62031)	.72390	.64194	.72618
a_7 Fitted	4.0700×10^6	4.0700×10^6	4.0700×10^6	4.0700×10^6
a_8 Fitted	-0.5	-0.5	-0.5	-0.5
a_9 Fitted	0.5	.48377	.49827	.48545
Avg Abs Err	-41.208 nW	.16012 nW	3.3773 nW	.20198 nW
SD Abs Err	68.673 nW	5.1505 nW	15.598 nW	5.2233 nW
Avg Rel Err	3.4327	-.057077	-.055610	-.063484
SD Rel Err	3.6250	.29058	.25896	.28383

Due to the number of constants fitted, auto-no-slack was used in Table 14, and has affected the fitting process by finding the a_7, a_8, and a_9 constant values closest to the theory values that create a good fit. As such, several constants exactly or closely match the theory values, indicating good theory versus data agreement or insufficient good data to pull those constant values away from the theory values. Only when a constant has changed significantly from its theoretical value would that show that the data required a different value to reduce the errors and obtain a good fit. As it turns out, the applied auto-no-slack shows that out of a_7, a_8, and a_9, the fitted constant a_9 was least able to maintain its theory value without causing a large increase in error for the fit. So a_9 was pushed slightly away from its theoretical value by the PIC data. Table 10 shows a similar trend in its fitted c' constant. Constant a_4 was also pushed slightly from its theory value when the auto-no-slack option set the a_8 constant to its theory value. The fitting program set a_8 to its theory value because a_8 had less affect on decreasing the fit error than a_4 and because only enough levels of β exist to fit one of the two constants. Finally, a_4 was reliably fitted in only two levels because there were two *sets* of points at the two β levels.

Now each of the fitted constants of Table 14 will be examined. For constants a_2 and a_3, there was only one level of PIC data sampled exploring modulation depth and illuminated spot area, and these level values are listed in the table captions. As such, these constants could not be fitted and were simply set to their theoretical values. Constants a_7, a_8, and a_9 were 'slack constants', remaining mostly unchanged from the theory values by the PIC data. Constants a_1 and a_6 were allowed to go to any value, since there were normalization questions about a_1, and a_6 had not been calculated from theory. Constants a_4 and a_5 were found to have varied from the theory values. As a consequence, the value of a_1 had to adjust itself to compensate. However, from a comparison viewpoint the constant differences that exist seem acceptable.

Besides the constants, how good are the overall fit errors? The standard deviation of the absolute errors varies from about 5 to 16 nW. To understand what that means and to put it into perspective, the smallest $\langle P_r \rangle_s$ measured was .076901 nW, while the largest was 352.91 nW. From the two resonance scans, the peak for the 0.2 eV curve was 0.48531 nW, while the peak for the 1.3305 eV curve was 78.197 nW. Of course these small $\langle P_r \rangle_s$ values are for a very small spot having an area of only 2.0×10^{-6} square meters. So, 16 nW is less than 5% of the maximum $\langle P_r \rangle_s$ value fitted, and less than 20% of the 1.3305 eV resonance curve maximum. For the standard deviation of the relative error, the values range from about 26% to 29%, and represent the relative error with respect to all the fitted points. This number means that while the data points varied widely from .076901 nW to 352.91 nW, the calculated values based on the theory equation using the fitted constants remained close in a relative sense to most of those points. This radiated power PIC data set was obtained precisely and should generally be considered relatively accurate over wide scales. This is especially true when fitting Tables 12-14 to all of the data set, with its huge variation in radiated power from .076901 nW to 352.91 nW. Again, when the fit errors are compared with the data, the errors are generally good given the simple theory and PIC code differences and limitations.

One last table I would like to present looks at a fit of the power radiated due to the self-oscillation of the emitted beam. In this case, the beam current density $\langle j\rangle$ has not been modulated to obtain resonance or off-resonance driven radiation. Table 15 presents the a' fit of $\langle P_r\rangle_s$, similar to Table 6, but for self-oscillation generated power. When Table 15 is compared with Table 6, where the only difference in the PIC simulations is that the Table 6 simulations were driven at resonance, the comparison shows that driving the simulations at resonance increased the radiated power about three to four times above the self-oscillation values. Also note that the fitted values of a' are close to the predicted theory values for the resonance-modulated radiated power. Since the theory is quite simple, *the theory may actually represent the self-oscillation radiated power, as the fits to the PIC simulation results presented in Table 15 (and the significance testing presented in Table 7) seem to suggest.* If so, the externally driven $\sin^2$ type of current density modulation and its affect on increasing the radiated power is not really modeled by the simple theory. Regardless, one of the main purposes of the simple theory was to extract scaling laws to compare with the PIC data, and for that purpose the scaling laws have performed well.

Table 15: a' Coefficient Fits and Errors of $\langle P_r\rangle_s = a'F_D^{d'}A^{e'}\beta^{b'}\langle j\rangle^{c'}$

Cost Func	a' Fitted	b'	c'	d'	e'	Abs Avg Err, nW	Abs SD Err, nW	Rel Avg Error	Rel SD Error
Theory	24.657	2	2	2	2	-2.2645	5.8101	.13219	.26137
F1	24.170	2	2	2	2	-3.0333	6.2424	.15500	.30385
F2	29.024	2	2	2	2	4.6279	13.858	-.038151	.21397
F3	24.926	2	2	2	2	-1.8393	6.5334	.11995	.27743
F4	22.999	2	2	2	2	-4.8823	6.9198	.21383	.35355
F5	32.633	2	2	2	2	10.325	22.457	-.14453	.24475

This section has demonstrated that the theory equations and theory scaling laws show good general agreement with the computational Particle-In-

Cell code simulations. The PIC code simulation results were ideally suited for comparison with the theory models as the PIC code implemented many of theory models directly and then simulated those models in a fully self-consistent, nonlinear manner. However, even with perfect nonlinear model simulation and perfect data, differences would be expected to arise from the predictions based solely on analytic, linear theory. Worse, much like in the laboratory, perfect data are hard to obtain. Hence, when the theory equations and PIC simulation data are compared, only general agreement can be expected at best, while large disagreement would tend to show up if the theory equations and PIC data did not generally agree. While inherent theory-versus-PIC disagreement was expected and in general significantly demonstrated by the tables, the differences seen seem reasonable based on the limited nature of the simple theory and data obtained via PIC code simulation. And it was shown that the theory equations did generally agree with the PIC results in sign and magnitude, and while this does not prove that the theory equations and PIC data agree or are correct, the general agreement provides satisfying additional confidence in the theory equations and scaling laws. In the end, the simple theory was able to predict the PIC code results to well within an order of magnitude on average from the PIC code simulation data values.

In summary, there are many potential sources of differences between the theory and PIC simulation results, some of which I have listed below.

1. A coarse grid was used in the PIC simulations, and preliminary PIC simulation studies seem to indicate that as the computational grid is made coarser (with an automatically recalculated, fixed percentage of the Courant condition, increase in the allowed time step), the postprocessor calculated radiated power increases, even though other simulation parameters are held constant.

2. The postprocessor calculated radiated power was calculated in the time-domain over all frequencies and then averaged in time, and does not represent only the $\nu_{\rm nat}$ frequency component of the total radiated power.

3. Since the theory is simple, the theory predictions for resonance radiated power $\langle P_r \rangle_s$ may actually represent the self-oscillation values calculated by the PIC code, as suggested by the curve fits of the PIC simulation results presented in Table 7 and by the Table 15 significance testing.

4. Postprocessor calculated radiation frequency and PIC code electron orbit parameters are hardly affected by changes in the computational grid coarseness.

5. Both the self-oscillation and resonance $E_{Z\ \mathrm{max}}$ (and d_{max}) values were similar on the PIC simulation graphs, and ν_{nat} is defined by the theory as the primary self-oscillation frequency.

6. The PIC code, while implementing many theory models exactly, implemented some theory models better, and some models differently such as the E_Z fields actually simulated by the PIC code runs including contributions from charge that initially left the simulation.

7. Even if the theory correctly modeled the PIC simulations, the simple analytic, linear theory results cannot be expected to exactly agree with the full nonlinear PIC simulation results, and the fitted scaling laws and constants would reflect any differences.

8. The errors of the fits generally reflect the errors estimated for the fitted data, as is the case for $E_{Z\ \mathrm{max}}$, d_{max}, and ν_{nat} with respect to oscillating observables and FFT box width considerations.

9. PIC simulation results can 'drift' when one to two million steps are used in the simulation, as was required for these PIC simulations.

10. If the PIC data and theory scaling laws did not generally agree, then substantial disagreement from the theory could easily cause much worse curve fits than were actually achieved.

Therefore, the results and comparisons presented in the tables should have and would show inherent differences when using careful significance testing unless all sources of 'error' are identified, quantified, and removed or included in the comparisons. Yet, with all of the potential sources of error, good general agreement between the theory and PIC code simulation results has still been demonstrated. Consequently, the theory equations and scaling laws are tentatively supported by the PIC simulation data that currently exist and appear to be useful for making predictions and for optimizing photoelectron-generated radiation.

Conclusions

The MAGIC PIC code simulations and subsequent radiation postprocessing produced a wealth of information. Information concerning the extent, dynamics, and field strengths of the electron sheath layer were generated. For typical problems, the electron sheath in steady state extended on the order of $d_{\mathrm{max}} \approx 10\ \mu\mathrm{m}$ from the conducting surface. Also, typical electric field strengths perpendicular to the conducting surface would peak at around $E_{Z\ \mathrm{max}} \approx 100$ kV/m near the conducting surface. Essentially one-dimensional, or small-spot, PIC code simulations of the electron sheath were required in order to effectively handle the diverse particle and field scales involved in simulating realistic, self-consistent problems. Postprocessing the PIC simulations' emitted current as a function of location and time provided realistic, far-field radiation results. The small-spot predictions for the average radiated power $\langle P_r \rangle_s$ from all PIC code calculated frequencies cannot easily be extrapolated to large-spot predictions where diffraction effects are significant. However, my theory paper [Ref. 1] showed that typical real metals illuminated with low-power, low-energy (less than 5.0 eV) photon sources have small photoelectron yields and are often limited to generating small-spot radiation. In any case, the theory, PIC simulation, and curve-fit results for $E_{Z\ \mathrm{max}}$, d_{max}, and ν_{nat} *can* be applied to understanding both small and large spots.

Many of the PIC code simulations were performed to explore the behavior of the radiation generated by the emitted electrons. The radiation lobe patterns were found to have the expected small-spot, dipole $\sin^2\theta$ shape. Further, for unmodulated, monoenergetic, and perpendicular electron emission, a self or naturally oscillating frequency ν_{nat} was found to arise for the emitted electrons, causing far-field radiation with that frequency. PIC simulations were used to generate two resonance curves of radiated energy versus ν_{mod}, by varying the current density sinusoidal modulation frequency ν_{mod} about the natural frequency ν_{nat}. Typical radiation frequencies simulated were around $\nu_{\mathrm{nat}} \approx 6$ GHz. The generated radiation's primary FFT frequency always matched the externally imposed current density modulation frequency ν_{mod}. As expected, the maximum radiation was generated when ν_{mod} was set equal to ν_{nat}, and was three to four times greater than the power radiated due to self-oscillation alone. However, modulation at frequencies higher than ν_{nat} was seen to cause decreases in radiated energy that can fall below the self-oscillation values. Equations were presented to facilitate converting PIC code simulated results into various quantities such as radiated power and radiation conversion efficiencies. The maximum emitted electron beam energy to radiated energy conversion efficiency η_{rsB} simulated was 8.5574×10^{-6}. Converted to η_{rsL}, the estimated maximum total light energy to radiated energy conversion efficiency simulated, using $\lambda_L = 532\times10^{-9}$ meters and $Y = 0.1$, was 4.8854×10^{-7}.

Finally, the theory scaling laws were analytically compared with the PIC simulation data using nonlinear curve-fitting techniques. **Appendix A** lists the PIC code data used for the curve-fit analysis. Curve fits of $E_{Z\ \mathrm{max}}$, d_{max}, ν_{nat}, and $\langle P_r\rangle_s$ were performed using the PIC data. The curve fits were performed with respect to the generalized theory equations for each observable, and were performed for one or more of the constants simultaneously. Several cost functions were used in the curve fits, and error statistics for each set of theoretical and fitted constants were generated.

While not perfect, the theory and data do generally agree. The simple theory and PIC code simulations were not expected to statistically match each other due to differences inherent in the theory and simulations. Yet the theory and the PIC data are well within an order of magnitude of each other in their predictions. The self-oscillation $\langle P_r\rangle_s$ and d_{max} fits were closest to the theory in significance testing and were close in magnitude as well, while the ν_{nat} fit was close in magnitude alone. The worst agreement came from the $E_{Z\ \mathrm{max}}$ and resonance $\langle P_r\rangle_s$ fits, which performed poorly in significance testing and were on average two to four times larger than the theory predictions. Where differences occurred, the differences seem acceptable due to the limited nature of the analytic solutions inherent in any simple theory equation when compared with the solutions obtained from the theory models as implemented in the MAGIC PIC code simulations. So from a comparison viewpoint the differences that exist in the theory versus fitted constant values, as well as the size of the fit errors, both appear acceptable, and the simple theory scaling laws and PIC simulation results generally agree and provide confidence in each other. As a consequence, the results presented in this paper indicate that the basic physics investigated is understood and that the simple analytic theory scaling laws perform, if anything, better than might be expected.

Appendix A

This appendix presents tabled summaries of the PIC code simulation data that were used to create various graphs and tables in this paper. *All* of the tabled data listed in this appendix were generated from MAGIC simulations performed on Sun workstations, and these were the only data used to perform the curve fits. However, Figs. 1-3 and 9-10 were missing graphs that were recreated by running MAGIC on one of the Los Alamos National Laboratory's Cray supercomputers, using the same decks that were originally run on the Sun workstations. Also, Cray runs of some of the same decks used on the Sun workstations, but with new data outputs, have been performed after the main body of this paper was written. Some of those Cray results have been summarized in **Appendix B**.

Table 16 presents the data from the self-oscillation PIC simulations for $E_{Z\ \max}$, $d_{\max}$, and $\nu_{\rm nat}$. Table 17 presents the both the unmodulated and resonance-modulated $\langle P_r \rangle_s$ data, in terms of the far-field energy radiated through a hemisphere, $E_{\rm rh}$. The conversion relationship used to convert radiated energy to average power was $\langle P_r \rangle_s = 2.0 E_{\rm rh}/t_{\rm run}$, where $t_{\rm run}$ was the total simulated run time of the PIC code. So both the self-oscillation and resonance $E_{\rm rh}$ data are listed for various PIC runs in Table 17. Also, note that while damped self-oscillation radiated energy data were included in Table 19 and were used in creating various graphs and Tables 12-14, the data from those runs were less representative of the runs that displayed steady-state self-oscillation. Consequently, the $\nu_{\rm nat}$ and $\langle P_r \rangle_s$ data from the damped self-oscillation PIC simulation were not used when fitting the smaller data sets shown in Tables 16-17. Tables 18-19 display the 48 nonresonance, or total radiated power data points, from which the 41 near-resonance $\langle P_r \rangle_s$ data points were selected to curve fit. Note that some of the data in Tables 18-19 repeat the data in Table 17, and that I have listed unmodulated run data from Table 17 as data with zero modulation frequency in Tables 18-19. So fits in Tables 3-6, 8-11, and 15 were performed using the PIC data in Tables 16-17. And fits in Tables 12-14 were performed using the *near*-resonance $\langle P_r \rangle_s$ data in Tables 18-19.

The independent parameters that were varied in each run include eV_b, $\langle j \rangle$, and $\nu_{\rm mod}$. The independent parameters that were constant for *all* of the PIC code simulations were $t_{\rm run} = 6.3263 \times 10^{-9}$ seconds, $D = 100\%$, and $A = 2.0 \times 10^{-6}$ m^2. The simulation grid was constant for *all* of the PIC code runs used in this paper, as was the calculated FFT box width, which was equal to 0.15815 GHz.

Table 16: PIC Code Simulation $E_{Z\ \mathrm{max}}$, d_{max}, and ν_{nat} Data

Data	eV_b	$2\langle j\rangle$	$E_{Z\ \mathrm{max}}$	d_{max}	ν_{nat}
Point	eV	$\mathrm{A/m^2}$	kV/m	μm	GHz
1	0.2000	2300.0	27	17.45	4.2700
2	1.3305	3444.0	58	29.83	damped osc
3	1.3305	7749.1	90	23.42	5.0608
4	1.3305	12000.0	unavailable	unavailable	6.0096
5	1.3305	13776.0	108	24.37	6.4841
6	1.3305	21525.0	152	19.32	7.9074
7	1.3305	30996.0	150	17.17	9.3308

Table 17: PIC Code Simulation Self-Osc and Res $\langle P_r\rangle_s$ Data

Data	eV_b	$2\langle j\rangle$	$E_{\mathrm{rh\ self-oscillation}}$	$E_{\mathrm{rh\ resonance}}$
Point	eV	$\mathrm{A/m^2}$	Joules	Joules
1	0.2000	2300.0	2.4625×10^{-19}	unavailable
2	1.3305	3444.0	damped osc	damped osc
3	1.3305	7749.1	3.3101×10^{-17}	1.1244×10^{-16}
4	1.3305	12000.0	7.7809×10^{-17}	2.4735×10^{-16}
5	1.3305	13776.0	1.0201×10^{-16}	3.1458×10^{-16}
6	1.3305	21525.0	2.0451×10^{-16}	6.4757×10^{-16}
7	1.3305	30996.0	3.6396×10^{-16}	1.1163×10^{-15}

Table 18: PIC Code Simulation 0.2000 eV $\langle P_r \rangle_s$ Data

Data Point	eV_b eV	$2\langle j \rangle$ A/m^2	ν_{mod} GHz	$E_{\mathrm{rh\ all}}$ Joules
1	0.2000	2300.0	0.0000	2.4625×10^{-19}
2	0.2000	2300.0	0.1000	2.4325×10^{-19}
3	0.2000	2300.0	0.5000	3.2776×10^{-19}
4	0.2000	2300.0	1.0000	5.8831×10^{-19}
5	0.2000	2300.0	2.0000	1.1155×10^{-18}
6	0.2000	2300.0	3.0000	1.4359×10^{-18}
7	0.2000	2300.0	4.0000	1.5351×10^{-18}
8	0.2000	2300.0	5.0000	1.4807×10^{-18}
9	0.2000	2300.0	6.0000	1.2841×10^{-18}
10	0.2000	2300.0	7.0000	1.1208×10^{-18}
11	0.2000	2300.0	8.5000	4.1302×10^{-19}
12	0.2000	2300.0	10.000	4.2274×10^{-19}
13	0.2000	2300.0	18.000	4.2839×10^{-19}
14	0.2000	2300.0	30.000	4.4951×10^{-19}

Table 19: PIC Code Simulation 1.3305 eV $\langle P_r \rangle_s$ Data

Data	eV_b	$2\langle j \rangle$	ν_{mod}	$E_{\text{rh all}}$
Point	eV	A/m^2	GHz	Joules
1	1.3305	3444.0	0.0000	1.3981×10^{-18}
2	1.3305	7749.1	0.0000	3.3101×10^{-17}
3	1.3305	13776.0	0.0000	1.0201×10^{-16}
4	1.3305	21525.0	0.0000	2.0451×10^{-16}
5	1.3305	30996.0	0.0000	3.6396×10^{-16}
6	1.3305	3444.0	6.0000	1.1322×10^{-17}
7	1.3305	7749.1	6.0000	1.0510×10^{-16}
8	1.3305	13776.0	6.0000	3.1429×10^{-16}
9	1.3305	21525.0	6.0000	6.0420×10^{-16}
10	1.3305	30996.0	6.0000	9.5021×10^{-16}
11	1.3305	3444.0	3.1630	2.9665×10^{-17}
12	1.3305	7749.1	5.0608	1.1244×10^{-16}
13	1.3305	13776.0	6.4841	3.1458×10^{-16}
14	1.3305	21525.0	7.9074	6.4757×10^{-16}
15	1.3305	30996.0	9.3308	1.1163×10^{-15}
16	1.3305	12000.0	0.0000	7.7809×10^{-17}
17	1.3305	12000.0	1.0000	1.1084×10^{-16}
18	1.3305	12000.0	2.0000	1.4501×10^{-16}
19	1.3305	12000.0	3.0000	2.0265×10^{-16}
20	1.3305	12000.0	4.0000	2.3492×10^{-16}
21	1.3305	12000.0	5.0000	2.2378×10^{-16}
22	1.3305	12000.0	6.0000	2.4735×10^{-16}
23	1.3305	12000.0	7.0000	2.3508×10^{-16}
24	1.3305	12000.0	8.0000	2.0502×10^{-16}
25	1.3305	12000.0	9.0000	1.6438×10^{-16}
26	1.3305	12000.0	10.000	1.4238×10^{-16}
27	1.3305	12000.0	11.000	1.4867×10^{-16}
28	1.3305	12000.0	12.000	1.1349×10^{-16}
29	1.3305	12000.0	13.000	3.5428×10^{-17}
30	1.3305	12000.0	14.000	3.9953×10^{-17}
31	1.3305	12000.0	15.000	6.0431×10^{-17}
32	1.3305	12000.0	16.000	8.0218×10^{-17}
33	1.3305	12000.0	17.000	8.6706×10^{-17}
34	1.3305	12000.0	18.000	8.7458×10^{-17}

Appendix B

This appendix presents a short summary of some MAGIC results that were simulated using a Cray computer at Los Alamos National Laboratory after the main paper was written. The 1.3305 eV self-oscillating MAGIC decks used for the Sun workstation simulations were modified to dump new and better output data, and simulations were performed. A l of the fit results listed in this appendix were calculated using the F3 cost function, and only involve fitting the a' constant to the self-oscillation data. Further, the amount of additional electric field strength due to the initial emitted charge escaping out of the simulation was precisely calculated.

The new curve-fit results are a, follows. The Cray PIC data confirmed that the average $E_{Z\ \mathrm{max}}$ fit is $a' = 27444$, with $\xi_{\mathrm{Asd}} = 1.30$ kV/m. These runs also found that for all of the 1.3305 eV simulations, the E_Z fields in the grid were increased by a *constant* 13.322 kV/m due to the escaped charge that was not returned to the conducting surface. When subtracted from the $E_{Z\ \mathrm{max}}$ data, this decreases the fitted value of a' for $E_{Z\ \mathrm{max}}$ by roughly 10%. Next, the Cray runs show that fitting d_{min}, using carefully dumped and sampled PIC data, results in $a' = 10.2$, with $\xi_{\mathrm{Asd}} = 1.1$ μm. For ν_{nat}, the fitted Cray PIC data give $a' = 3.555 \times 10^6$, with $\xi_{\mathrm{Asd}} = 0.137$ GHz. Finally, for $\langle P_r \rangle_s$ the fitted Cray PIC data give $a' = 25.08$, with $\xi_{\mathrm{Asd}} = 7.60$ nW.

Appendix C

This appendix discusses how to do useful, experimentally realistic, two-dimensional PIC code simulations, using existing PIC codes. As discussed at the beginning of this paper, it would normally take about a million time steps and a million grid cells to do a realistic simulation of an actual two-dimensional experiment using standard PIC codes. However, there is a way to study realistic two-dimensional problems and physics *by scaling the problem parameters.* To do so, *scaling runs can be designed* to allow PIC code simulations to be performed using the capabilities of existing PIC codes. The results of these scaling runs can then be adjusted to the unscaled parameter settings to determine the results in the region of parameter space of interest. This scaling process is greatly facilitated because theoretical scaling laws exist that can be used to reduce the number of PIC code runs necessary to design the scaled simulations and to unscale the results.

However, scaling is something of an art. Scaling involves identifying all the necessary parameters and understanding the geometry, scales, and physics of the problem. And the researcher must know why a PIC code is limited and cannot perform the simulations as the problem specifies. For this problem, the scale lengths of the electron trajectories versus the size of the problem, the electromagnetic propagation speeds, and the size of the generated radiation are basically too diverse, so the PIC simulation can not readily solve the real problem directly. The researcher must then change selected parameters so as to allow performing a valid two-dimensional simulation that preserves all or a significant part of the physics being investigated. The scaling process can be further complicated by various parameters that are linked, so that desired changes in the scaling of one parameter may create undesired changes in another. However, the process of scaling these low-energy photoelectron problems is made easier by having an analytic theory, one-dimensional PIC simulation results, and experience with PIC codes. Lastly, for these problems and scaling laws, tests will need to be performed in one and two dimensions to confirm that the scaling laws actually work and apply to the scaled results. But once the above considerations are satisfied, one can feel confident that the scaling performed applies to the physics being investigated.

APPENDIX C

Below, I list some general scaling rules of thumb to try and satisfy as much as possible. Generally,

1. Keep as close to the original parameters and scaling of interest when possible.

2. Keep significant relative geometries and ratios.

3. Keep relevant characteristic spatial and temporal scales and effects.

For these specific photoelectron problems, the parameters could be easily scaled by either increasing the frequency or energy of the problems. Either scaling will allow 2-D runs looking at realistic area effects. Frequency scaling requires simultaneous scaling increases in ν_{nat}, β, and $\langle j \rangle$ while maintaining realistic electron trajectories on the order of microns, and correspondingly small PIC simulation time steps (roughly, grid cell sizes of 1 micron in z, by 3 to 5 microns in r). Fortunately, the smaller wavelength requires fewer radial cells (such as with respect to scaled diffraction, for instance), and fewer time steps are needed to simulate 10 high frequency cycles that roughly would be on the order of 300 GHz to 3000 GHz for a 1 mm to 100 μm wavelength. Essentially, with frequency scaling the grid size and number of cells shrinks, as does the number of time steps necessary to run 10 cycles. Energy scaling requires scaling increases in β, $\langle j \rangle$, and d_{max}, thereby significantly increasing the size of the PIC simulation time step, while maintaining both the realistic problem frequencies (1 to 10 GHz) and realistic problem radial spot sizes. Essentially, with energy scaling the larger grid cell size requires fewer grid cells to reproduce realistic radial spot sizes, and also increases the size of the time steps, so that the number of time steps needed to simulate a few realistic ν_{nat} cycles decreases. I have chosen to present energy scaling since I am more interested in maintaining the original parameters for frequency and spot size than for realistic electron trajectories.

For energy scaling of the specific problem,

1. Want to do two-dimensional area simulations to look at area diffraction, more realistic current distributions in space and time, realistic two-dimensional sized and shaped structures, all solved self-consistently, and want to be able to reproduce any two-dimensional experimental data that might exist.

2. Keep simulations classical, like the original problem, so that $\beta \leq 0.1$ or $\beta \leq 0.3$ at most.

3. Even with energy scaling, keep $d_{\mathrm{max}} \ll \lambda_{\mathrm{rf}} \ll R$, like the original problem, and to satisfy this model constraint that was used in developing the theory radiation equations. Note that $\lambda_{\mathrm{rf}} = \lambda_{\mathrm{mod}}$, and that $\lambda_{\mathrm{mod}} = \lambda_{\mathrm{nat}}$ to satisfy the steady-state resonance condition. Then

$$1 \ll \frac{\lambda_{nat}}{d_{\mathrm{max}}} \ll \frac{R}{d_{\mathrm{max}}} \tag{45}$$

where the second condition can always be satisfied by moving the observer accordingly.

4. Keep $1 \leq \nu_{\mathrm{mod}} \leq 10$ GHz, as for the original problem.

5. Keep $1 \leq \nu_{\mathrm{nat}} \leq 10$ GHz, as for the original problem.

6. Use a grid size $z_{\mathrm{max}} \approx \lambda_{\mathrm{nat}}$, where the electron sheath is only simulated, and the radiation must be postprocessed similar to the 1-D PIC results presented in this paper.

where for the PIC simulations, the scale lengths for the particles and fields are now closer together, allowing two-dimensional problems with larger time steps while maintaining good spatial and temporal resolutions while increasing β, $\langle j \rangle$, and the z-axis cell size.

To roughly show how the energy scaling would work, I will now discuss two examples using energy scaling to construct workable two-dimensional, cylindrical, axial symmetric, PIC code simulations. These examples illustrate techniques that can be used to set up other two-dimensional scaled PIC code simulations involving similar low-energy photoelectron problems while demonstrating the typical simulation designs and run times that can be achieved by scaling. The first example will be for emitted photoelectrons scaled to an energy such that $\beta = 0.1$, and the second example will scale the problem to a approximate maximum case where $\beta = 0.3$. The larger that β gets, the easier the PIC simulation is to perform, and the further the problem becomes scaled from the original problem parameters. The other independent parameters selected will be λ_{nat} and t_{run}. From β, λ_{nat}, and t_{run}

the scaling PIC code simulation parameters are constructed. The equations that are used are

$$
\begin{aligned}
E_Z &= a_E \beta^{1/2} \langle j \rangle^{1/2} && (46)\\
d_{\mathrm{max}} &= a_d \beta^{3/2} \langle j \rangle^{-1/2} && (47)\\
\nu_{\mathrm{nat}} &= a_\nu \beta^{-1/2} \langle j \rangle^{1/2} && (48)\\
\langle P_r \rangle_s &= a_p F_D^2 A^2 \beta^2 \langle j \rangle^2 F_R && (49)\\
c &= \lambda_{\mathrm{nat}} \nu_{\mathrm{nat}} && (50)\\
\gamma &= (1 - \beta^2)^{-1/2} && (51)\\
eV_b &= 511000(\gamma - 1) && (52)\\
\frac{\lambda_{\mathrm{nat}}}{d_{\mathrm{max}}} &= \frac{c}{a_\nu a_d} \frac{1}{\beta} && (53)\\
&= \frac{4}{\beta} \quad \text{theory} && (54)\\
&\approx \frac{4.99}{\beta} \quad \text{PIC F3 fit} && (55)\\
\langle j \rangle &= \frac{c^2}{a_\nu^2} \frac{\beta}{\lambda_{\mathrm{nat}}^2} && (56)\\
&= \frac{2^2 \epsilon_0 m_e c^3}{e} \frac{\beta}{\lambda_{\mathrm{nat}}^2} \approx 5425.6 \frac{\beta}{\lambda_{\mathrm{nat}}^2} \quad \text{theory} && (57)\\
&\approx 6590 \frac{\beta}{\lambda_{\mathrm{nat}}^2} \quad \text{PIC F3 fit} && (58)\\
\Delta z &\approx d_{\mathrm{max}}/15 && (59)\\
0.2\Delta z &\leq \Delta r \leq 5.0\Delta z && (60)\\
z_{\mathrm{min}} &\equiv 0.0 && (61)\\
z_{\mathrm{max}} &\approx \lambda_{\mathrm{nat}} && (62)\\
r_{\mathrm{min}} &\equiv 0.0 && (63)\\
r_{\mathrm{max}} &\approx \lambda_{\mathrm{nat}}/2 && (64)\\
z \text{ cells} &\approx z_{\mathrm{max}}/\Delta z && (65)\\
r \text{ cells} &\approx r_{\mathrm{max}}/\Delta r && (66)\\
\Delta t_{\mathrm{run}} &< \frac{1}{c} \left[\frac{1}{(\Delta r)^2} + \frac{1}{(\Delta z)^2} \right]^{-1/2} && (67)\\
\text{timesteps} &= t_{\mathrm{run}}/\Delta t_{\mathrm{run}} && (68)
\end{aligned}
$$

where the coefficients can be selected from theory or from various cost functions for the a' only fits to the PIC data. Also, the power equation is for testing 1-D scaling at high energies, and if it holds, then a similar, modified, generalized power equation could be fitted to the large-spot PIC code data and scaled back to lower energies. Finally, the number of cells depends on various nonuniform cell strategies used to speed up the PIC code simulation versus a uniform cell size in each dimension.

Table 20 lists the various results for setting up energy scaled PIC code simulations for examples 1 and 2, using the F3 cost function's a' fitted coefficient values in the scaling calculations. So example 1 would require 40000 time steps and 250 by 500 cells, which is a medium size, doable, two-dimensional problem. Even better, example 2 would require 11800 time steps and 75 by 200 cells, which is an easy, fast, two-dimensional PIC code simulation of the problem.

Preliminary PIC code simulation 1-D scaling runs were performed using Sun workstations to test the scaling laws at predicting PIC code results at emitted electron energies up to a maximum value of $\beta = 0.3$. It should be noted that the equations used to do the far-field radiation postprocessing should work with classical and relativistic particles, as long as the PIC code correctly handles the simulation. For these runs, relativistic particle treatment was turned on in the MAGIC code as a precaution in case relativistic effects began to noticeably affect the results, though later tests showed that relativistic particle treatment was unnecessary as differences arising from simulation results using nonrelativistic particle treatment were not significant. To do the runs, an expert system MAGIC deck was set up that would automatically generate the MAGIC simulation grid and outputs with appropriate spatial and temporal resolutions, similar to the more simple worked examples shown in Table 20, for various inputs of β, $\nu_{\text{mod}} = \nu_{\text{nat}}$, and t_{run}. Data were not collected to test the scaling laws for E_Z with these PIC simulations. But the runs showed that *the scaling laws worked exceptionally well* all the way up to the maximum value of $\beta = 0.3$ tested for predicting d_{max} and ν_{nat}. In addition, it was also found that these observables were very insensitive to changing spatial and temporal resolutions tests that were easy to perform by changing the rules for those resolutions in the automatic MAGIC input deck. However, $\langle P_r \rangle_s$ was found to be sensitive to resolution changes, and the few simulations performed showed that as the grid became coarser, the value of the postprocessed radiation increased. The effect of spatial and

Table 20: Two Example PIC Code Energy Scaling Problems

Parameter	Units	Relation	Example 1	Example 2
β		$\equiv$	0.1	0.3
ν_{nat}	GHz	$\equiv$	3.0	3.0
t_{run}	ns	$\equiv$	10.0	10.0
γ		$=$	1.0050378	1.048285
eV_b	eV	$=$	2574.3	24674
$\lambda_{\text{nat}}/d_{\text{max}}$		$=$	49.9	16.6
λ_{nat}	mm	$=$	100.0	100.0
d_{max}	mm	$=$	2.01	6.02
$\langle j \rangle$	A/m^2	$=$	65900	198000
Δz	mm	$\approx$	0.1	0.33333
Δr	mm	$=$		0.66666
z_{min}	mm	$\equiv$		0.0
z_{max}	mm	$=$	100.0	100.0
r_{min}	mm	$\equiv$	0.0	0.0
r_{max}	mm	$=$	50.0	50.0
z cells		$=$	50	45
at Δz	mm	$=$	0.1	0.33333
z cells		$=$	450	155
at Δz	mm	$=$	0.1 to 0.322	0.33333 to 0.76344
r cells		$=$	250	75
at Δr	mm	$=$	0.2	0.66666
Δt_{run}	ps	$\leq$	0.29834	0.99450
Δt_{run}	ps	$\equiv$	0.25	0.85
time steps		$=$	40000	11800
z cells total		$=$	500	200
r cells total		$=$	250	75

temporal resolutions in calculating the radiation must be understood to do the scaling runs. Since the PIC simulation results in the rest of the paper were at fixed spatial and temporal resolutions for all of the runs, the scaling laws should generally capture the scaling in the parameters, with any major problems from increased radiation primarily affecting the fitted a' coefficients with a systematic error showing up as generally larger or smaller values.

In conclusion, scaling can be used to perform two-dimensional simulations of the problem using standard PIC codes, and energy scaling seems best in my opinion. The results solve the basic nonlinear problem self-consistently. Due to 2-D symmetry constraints, cylindrical problems will generally need to be illuminated perpendicularly, whereas Cartesian problems can be illuminated at an angle. Otherwise, realistic problems can begin to be simulated in two dimensions. PIC code modified theoretical scaling laws were tested and found to be useful in designing scaled PIC code simulations and in unscaling the PIC code results for the observables tested. One problem that must still be addressed is the effect of grid resolution on the predicted radiation, and what maximum grid resolution can be used while giving consistent radiation predictions at various emitted electron energies. Still, manual scaling involving more computer runs and little reliance on the scaling laws could still be performed in one and two dimensions to scale and unscale the problems. In summary then, carefully designed, scaled PIC code simulations are a doable and powerful way of doing self-consistent, two-dimensional PIC code simulations exploring experimentally realistic and interesting problems involving the photoelectron sheath behavior, generated field strengths, and generated far-field radiation.

References

[Ref. 1] Thomas Dipp. *A Simple Model for Determining Photoelectron-Generated Radiation Scaling Laws.* Los Alamos National Laboratory Technical Report LA-12674. Los Alamos National Laboratory Pulsed Energy and Diagnostic Physics Group (X-5), Los Alamos, New Mexico, 1993.

[Ref. 2] Bruce Goplen, Larry Ludeking, Gary Warren and Richard Worl. *MAGIC User's Manual.* Mission Research Corporation Technical Report MRC/WDC-R-246. Mission Research Corporation, Newington, VA, 1990.

[Ref. 3] Thomas A. Oliphant, Jr., MERLIN Code Developer. Personal Consultations. Los Alamos National Laboratory Pulsed Energy and Diagnostic Physics Group (X-5), Los Alamos, New Mexico, 1992.

[Ref. 4] Philip R. Bevington. *Data Reduction and Error Analysis for the Physical Sciences.* USA: McGraw-Hill, Inc., 1969.

[Ref. 5] Richard L. Branham, Jr. *Scientific Data Analysis: An introduction to Overdetermined Systems.* New York: Springer-Verlag New York Inc., 1990.

END

DATE
FILMED
3/30/94