Beam Emittance in the ERL Injector Prototype

Peter Battaglino
Physics Department, Carnegie Mellon University, Pittsburgh, PA, 15213
(Dated: August 8, 2003)

The number of parameters which influence the beam emittance produced by the injector for the Cornell ERL is very large. Computer simulations to determine the parameter set that optimizes the emittance may easily become so large as to be intractable. We have developed a framework for large simulation runs, allowing variation of a reasonable parameter set over realistic ranges. The supercomputer `Feynman` was utilized for its multiprocessing capabilities. A large initial simulation has been completed, establishing the utility of this scheme. In addition, we have established empirical relationships between the minimum emittance, the bunch length and the bunch charge for the particular injector configuration modeled.

I. INTRODUCTION

The injector for Cornell's proposed Energy Recovery Linac (ERL)[1] must provide a beam of very small size and low divergence to the main linac if we are to expect X-rays for the Cornell High Energy Synchrotron Source (CHESS) to come close to anticipated coherence – a very desirable characteristic for many areas of X-ray physics. The size and divergence of the beam are qualities that are both summed up nicely in the *emittance* of the beam. Beam emittance is given approximately by the product of spot size and divergence, and is proportional to the volume of the bounding ellipsoid that each bunch occupies in its region of six-dimensional phase space. Each point in this phase space is characterized by a phase space vector

$$\mathbf{r} = (x, y, z, p_x, p_y, p_z). \tag{1}$$

In many cases the physics of the problem allows us to separate this six dimensional space into three decoupled two dimensional spaces. Additionally, we can recognize that the momentum of each electron is mostly longitudinal, so we can further simplify matters by considering, at least in the transverse case, the angle that the particle's momentum vector projected onto the x-z plane (in the case of x-emittance) makes with the longitudinal axis, instead of p_x. We denote this angle by x' (or y' in the case of y-emittance). Now we can express the transverse coordinates of a point in phase space as:

$$\mathbf{r}_x = (x, x'), \qquad \mathbf{r}_y = (y, y'). \tag{2}$$

The rms of, for example, the x-emittance of the bunch is given by:

$$\epsilon_{rms} = \sqrt{\langle x^2\rangle \langle x'^2\rangle - \langle xx'\rangle^2}. \tag{3}$$

One last consideration to make is that of normalization of emittance. For instance, if we accelerate a bunch, we are increasing each electron's longitudinal momentum without increasing transverse momentum. As a result of this, the corresponding transverse angles are going to shrink for each electron, and total emittance will decrease. In order to account for this effect, we often use a quantity called the *normalized emittance*, which is defined as

$$\epsilon_{\text{norm},x} = \gamma\beta\epsilon_x \tag{4}$$

where $\gamma = 1/\sqrt{1-\beta^2}$ and $\beta = v/c$.

We would like to be able to achieve low emittances for a wide range of bunch charge, e.g. from 7.7 pC to 77 pC, determine a relation between the bunch charge and the minimum achievable emittance, and obtain an understanding of other factors that influence minimum emittance. The optimization of the injector is a very complicated process for several reasons. One main complexity can be attributed to the non-negligible effect of space charge. The outward force on an electron in a continuous, uniform, cylindrical beam can be shown, by summing the induced magnetic force caused by the moving beam and the electrostatic force to be

$$\mathbf{F}_{\mathrm{net}} = \frac{1}{\gamma^2}\frac{\rho_0 e r}{2\epsilon_0}\hat{\mathbf{r}}. \tag{5}$$

Thus at ultra-relativistic beam energy, the space charge forces and magnetic forces exactly cancel. Through most of the injector, however, γ is not very large, and the effects of space charge can dominate electron bunch behavior.

For this reason, we use the computer program ASTRA (**A** **S**pace Charge **Tr**acking **A**lgorithm) to simulate bunch evolution through the injector[1]. ASTRA allows us to simulate different configurations for a wide range of injector parameters such as charge per bunch, gun voltage, initial beam spot size at the cathode, solenoid and cavity field strengths, RF cavity phase angles, etc. Finding an emittance minimum in this multi-dimensional parameter space quickly becomes computationally difficult, and a full mapping of sizeable sections of this space cannot be completed on a standard computer in any reasonable amount of time. For example, if we were to vary six parameters independently, each having ten steps, we would be faced with the challenge of completing a million simulations. At one minute per simulation, this would take almost seven hundred days, or a little less than two years. However, supposing we ran 120 simulations at a time, this would be reduced to less than a week. It is therefore crucial that we not only provide a framework for automating a large number of simulation runs, but that we also find a way to break the task into several pieces and execute them simultaneously.

II. INJECTOR SIMULATION PARAMETERS

Fig. 1 illustrates the layout of the injector[2]. Electron bunches are produced at a GaAs photo-cathode illuminated by a 1300 MHz optical pulse train from a Ti:sapphire laser. In our simulations, initial rms bunch durations will be set to 20 ps with a flat-top radial distribution and a gaussian longitudinal distribution. The gun will be operating between 500 and 750 kV, and will have a 20° cathode electrode cone angle for initial beam focusing[3]. A solenoid is located about 29 cm from the photo-cathode for focusing, followed by a single cell buncher cavity at about 80 cm. Immediately following the buncher is another solenoid, followed by five two cell SRF cavities. Injector energies should range from about 5 to 15 MeV. In all simulations presented, thermal emittance is taken into account, where thermal emittance is given by the relation

$$\epsilon_{th} = \sigma_r\sqrt{\frac{E_{th}}{m_e c^2}}, \tag{6}$$

[1] Obtainable from the web at http://www.desy.de/~piot/Simulation/Astra_Released_Linux_Static.dir/

where E_{th} is dependent upon the photo-emitter material, the temperature, and the wavelength of the incident laser light. For GaAs at room temperature and a laser wavelength of about 780 nm, this value is 35 meV, and for a σ_r of 1 mm, corresponds to a ϵ_{th} of 0.2617 mm-mrad.

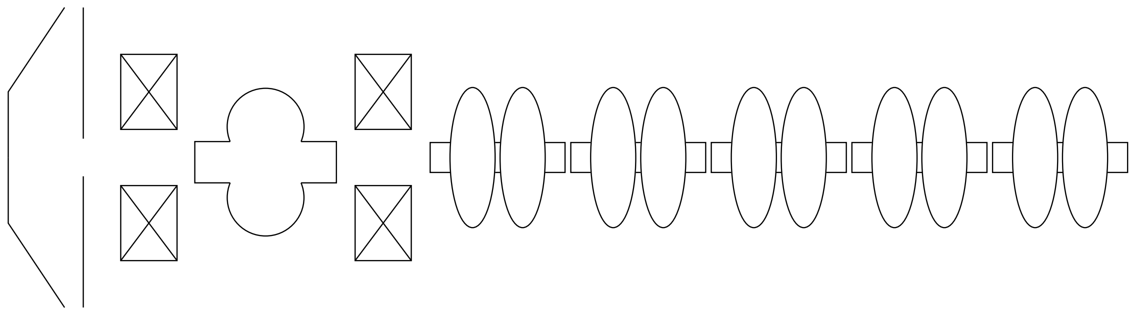

FIG. 1: Schematic of injector layout used in simulations.

Injector simulations are run with 1000-macroparticle distributions to allow for faster processing (a realistic number of electrons per bunch would be between 10^8 and 10^9). ASTRA forms these macroparticles by slicing the bunch in the cylindrical z and r coordinates and letting each resultant cylindrical annulus represent a macroparticle. The results of two ASTRA simulations are plotted in Fig. 2, where the field strength of the first solenoid was varied between 0.044 T and 0.051 T, and that of the buncher cavity was varied between 1.35 MV/m and 1.30 MV/m.

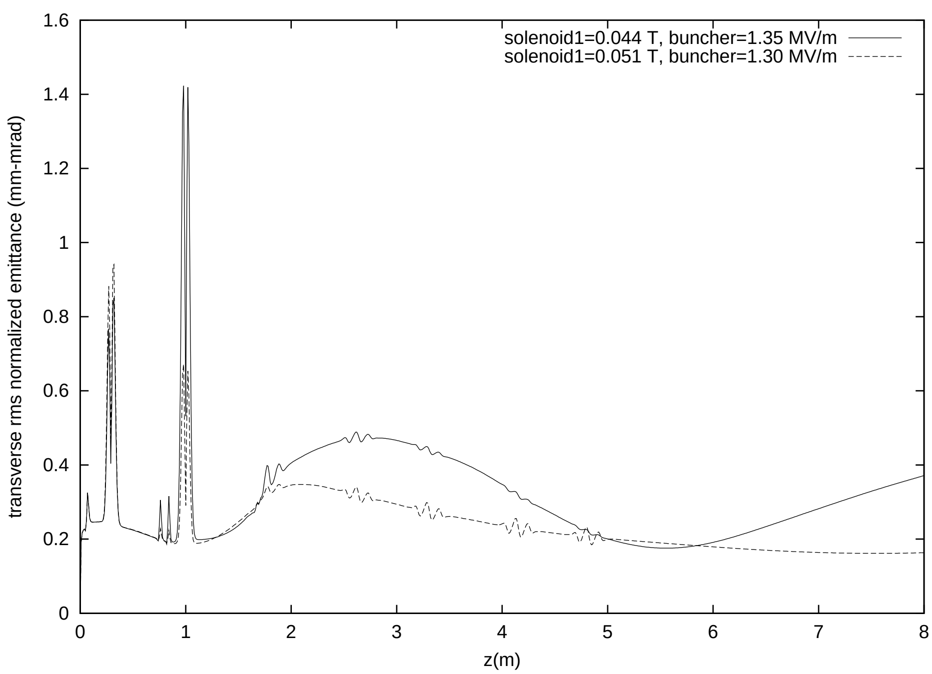

FIG. 2: Result of changing strength of first solenoid and buncher.

It appears that by changing these two values we have improved the overall performance of our injector. However, one must be aware not only of how the transverse emittance changes, but also of how the bunch length changes. For instance, one could imagine that decreasing transverse emittance might squeeze the bunch out longitudinally. *I.e.,* we may simply be transferring emittance from the transverse dimensions into the longitudinal dimension. This can have negative effects on beam energy spread due to the time-varying nature of the SRF accelerating fields: if the bunch length is too long, then different regions of the bunch will be accelerated differently, resulting in a smearing of the bunch energy in the longitudinal coordinate. Fig. 3 shows the effects of the changes in Fig. 2 on the length of the bunch.

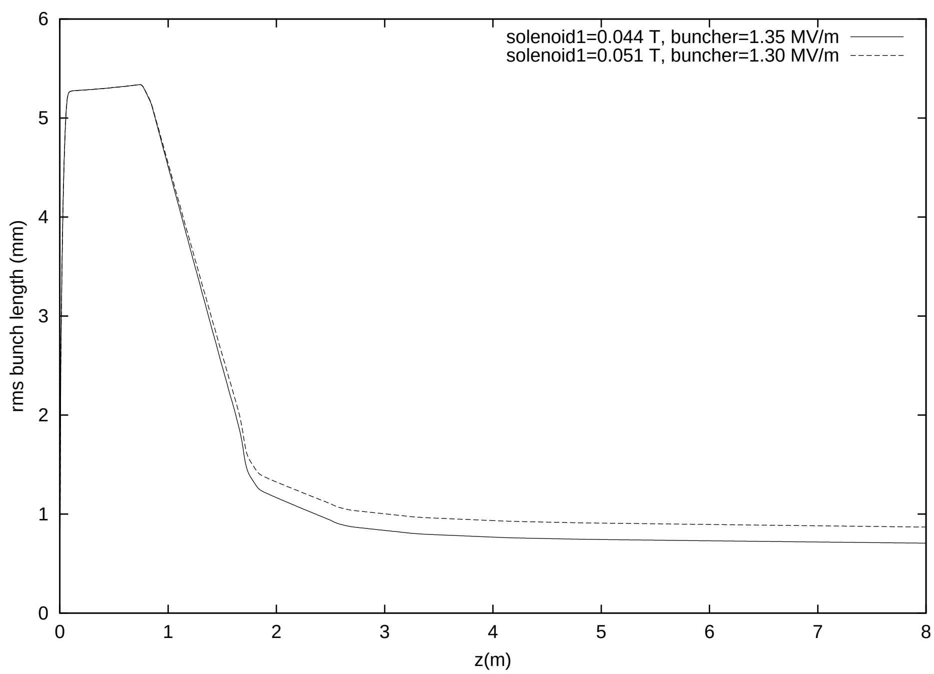

FIG. 3: Effects of transverse emittance reduction on bunch length.

As space charge effects influence bunch behavior so strongly, it is important to understand how bunch characteristics change with total charge per bunch. Fig. 4 shows the values of several bunch attributes at the end of the injector as a function of the total charge per bunch.

The general increasing nature of each of the curves in Fig. 4 shows what we would expect: the beam has more of a tendency to diverge and become less bright as we increase the bunch charge. However, the shapes of these curves are all likely to change significantly as we vary other parameters such as initial spot size, field strengths, cavity phases, etc. For instance, if we decrease spot size from its value of 0.4 mm in the plots of Fig. 4 to something smaller, like 0.1 mm, we might expect that several of the curves would blow up to much higher values at a much faster rate as charge is increased. Fig. 5 shows the result of this simple change in initial spot size.

From Fig. 5 it is evident that changes in just one parameter can correspond to dramatic changes not only in scaling factors of bunch characteristics but also in their functional forms. It is for exactly this reason that we must consider such a broad parameter space in its entirety and not just piece together our understanding from subspaces thereof.

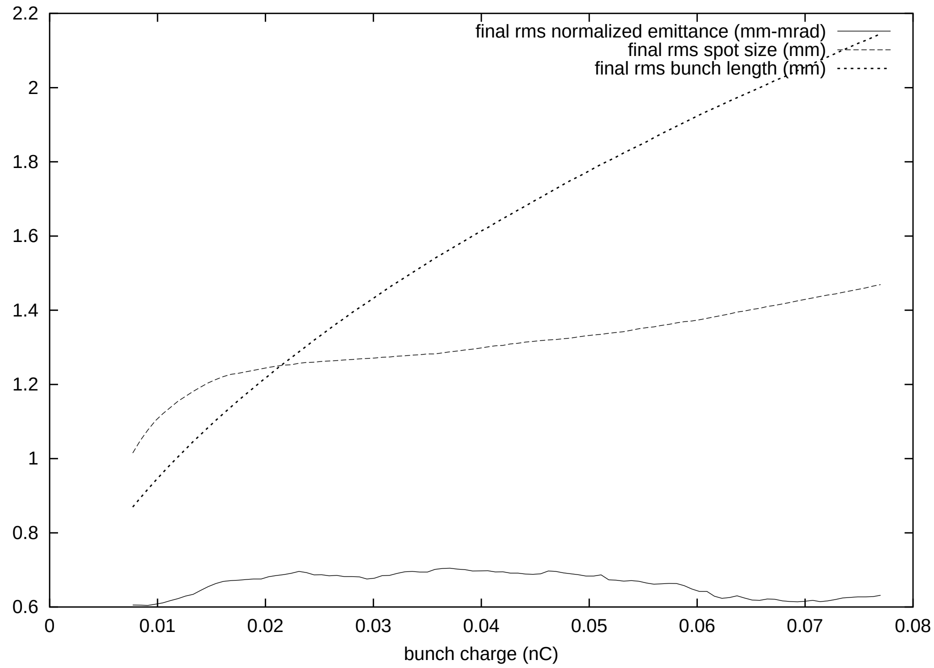

FIG. 4: Transverse emittance, spot size and bunch length as a function of total bunch charge with initial $\sigma_{r,\mathrm{rms}}$ of 0.46 mm.

III. AUTOMATED INJECTOR SIMULATION

The programming language Java was used to create a program that reads in a user-specified list of injector parameters, minimum and maximum values for them, and the number of steps for each. A set of all input files for ASTRA is then created, and ASTRA is run on each of them. Final emittance values are appended to a file. A graphical description is shown in Fig. 6.

This program functions well for small numbers of simulations (up to one thousand). As the need for larger data sets increases, so does the need for computational power: An attempt was made at interfacing the pre-existing Java program with the CHESS supercomputer at Cornell, but the way Java handled separate process branching with large amounts of simulations prevented success. Instead, a linux shell script was written that steps recursively over our N-dimensional parameter space, generating ASTRA input files from parameter values on the fly. In addition, a framework was integrated into the script to allow all simulations to be done on the CHESS supercomputer, which is a supercomputing cluster with sixty-four usable nodes and two processors per node. A diagram of the script-supercomputer setup is given in Fig. 7.

With this new method of completing simulations, sets of data with as many as two hundred thousand points became a matter of two days simulation time.

Ivan Bazarov built upon this script in C++ to constrain the regions over which we scan to those where emittance is below a certain tolerance value. The algorithm picks a successful seed point (*i.e.,* one whose corresponding emittance value is less than the specified maximum) and creates a box of $3^N - 1$ points around it, evaluating the emittances for each. If one of them is above the maximum emittance, that point is left out of further iterations. All other successful points are treated in the same manner as the first seed point. It is possible for

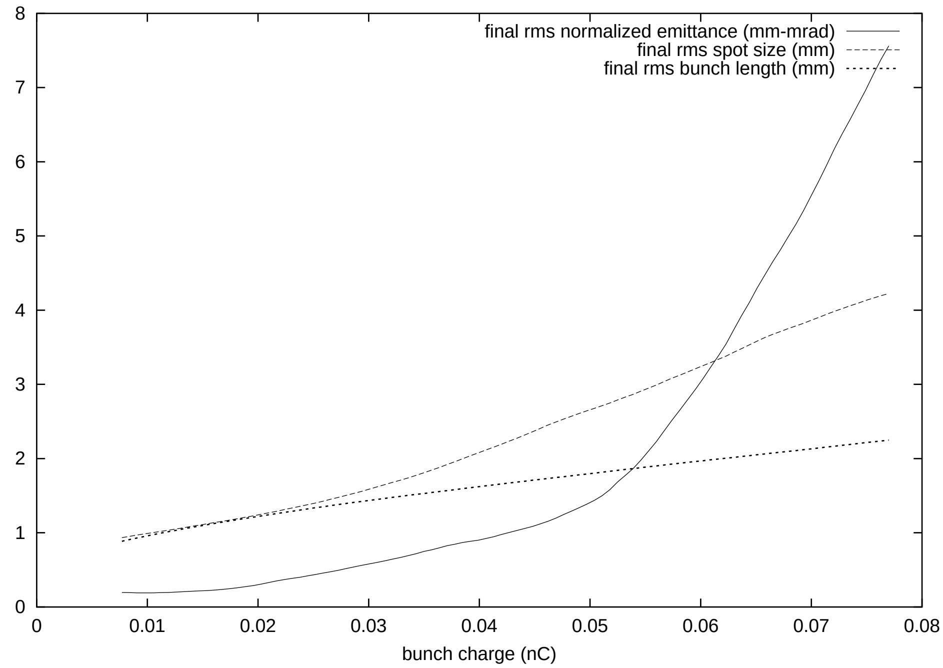

FIG. 5: Transverse emittance, spot size and bunch length as a function of total bunch charge with initial $\sigma_{r,\mathrm{rms}}$ of 0.2 mm.

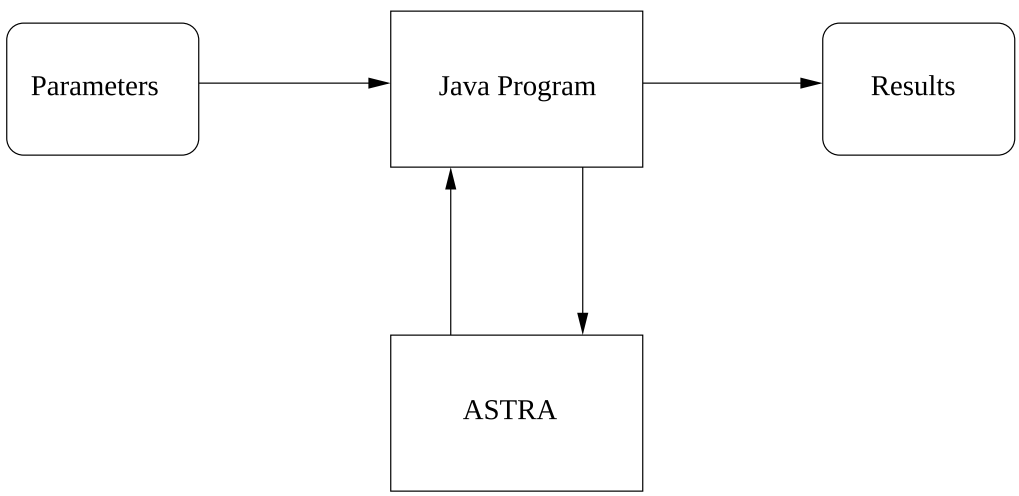

FIG. 6: Layout for Java simulation manager.

the algorithm to cover all feasible subsets of our initial region as long as there are no 'walled in' local minima — *i.e.,* as long as there is no $(N-1)$-dimensional boundary around such a subregion along which emittance values exceed the maximum. Such a situation appears most unlikely based on our current perception of the geometry of the surface that the injector's emittance traces out in N dimensions.

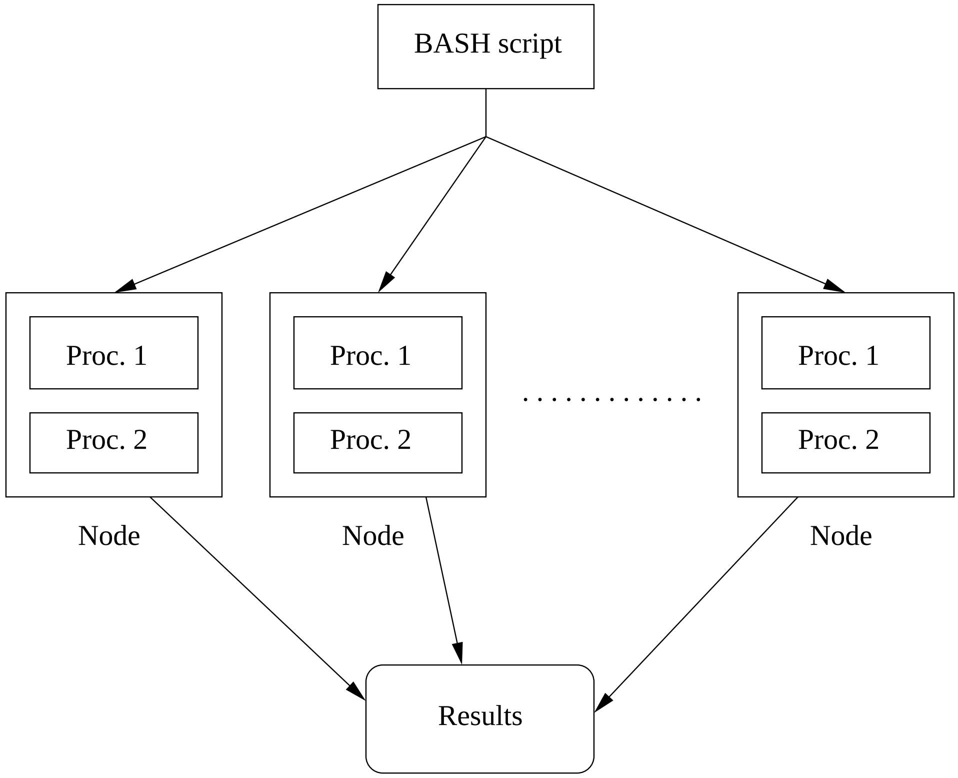

FIG. 7: Layout for supercomputer interface.

IV. RESULTS

Fig. 8 shows a scan over charge per bunch and the field strength of the first solenoid, where the simulations were truncated to finish in the middle of the buncher. From this plot it is evident that even a two-parameter scan can yield structure in the emittance surfaces. The existence of troughs such as that in Fig. 8 is encouraging in that it suggests the possibility of low emittances for a large range of bunch charges if we stay within that trough.

The CHESS supercomputing cluster was used to compute evolved bunch properties at the end of the injector (at 8.00 m) for the parameters and ranges shown in Table I.

TABLE I: The parameter values used for a general survey of our parameter space.

Parameter	Minimum	Maximum	Steps
Initial spot size (mm)	0.1	1.0	6
Solenoid field strength #1 (mT)	42.5	60	7
Solenoid field strength #2 (mT)	20	42.5	9
Buncher field strength (MV/m)	1.3	1.7	10
SRF cavity phase offsets (°)	-15	-2	5
Bunch charge (pC)	7.7	111.65	7

Fig. 9 is a plot of the minimum attainable emittance in the region of the parameter space we considered as a function of bunch charge.

The relation shown in this plot indicates a fairly linear correspondence between bunch

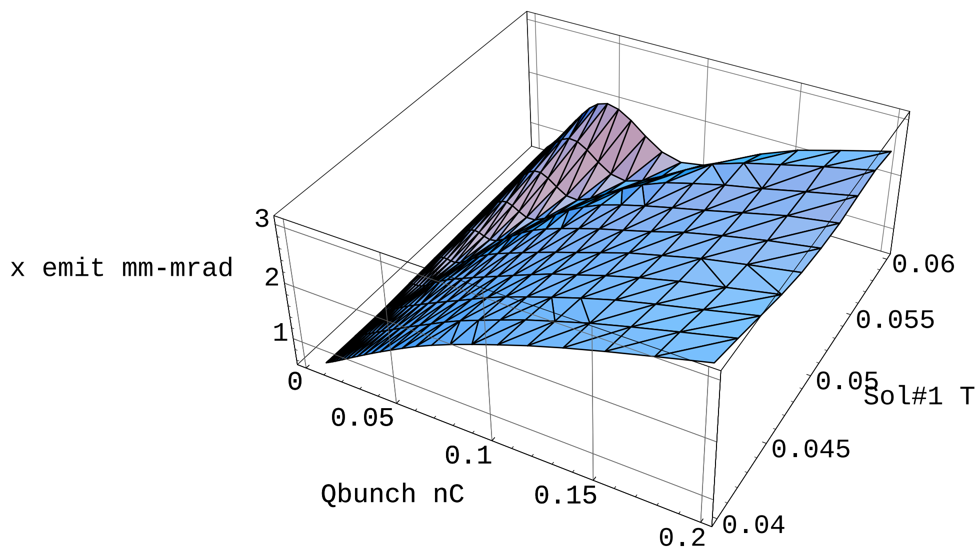

FIG. 8: Transverse emittance as a function of bunch charge and solenoid field.

charge and minimum emittance, with an rms residual of 0.029 mm-mrad. As Fig. 3 shows, the transverse emittance does not give us all of the information we need: we must also examine the bunch lengths that correspond to each minimum emittance value. These are plotted in Fig. 10.

Thus we see that if we limit our bunch length to a range of values, we will get a corresponding range of minimum possible emittance values. A scatter plot of this relation is shown in Fig. 11.

From Fig. 11 we can see that as we push transverse emittance down further and further, the bunch squeezes out into the longitudinal direction. This effect is also amplified by what appears to be an increasing, linear dependence on bunch charge. This combined relation might be characterized as

$$\epsilon_{\rm min} \sim \frac{Q_{\rm bunch}}{\sigma_{z,\rm max}^{p}}, \qquad p > 0. \tag{7}$$

Table II shows the results for absolute minimum emittance for 7.7 pC and 77 pC bunch charges. The parenthetical set of values describes a configuration that has an emittance slightly above minimum for that charge, but a considerably lower bunch length than that of the minimum emittance configuration.

V. CONCLUSIONS

A foundation has been laid for large-scale simulation runs to map out an N-dimensional parameter space and to explore beam parameters such as emittance in a quick, computation-

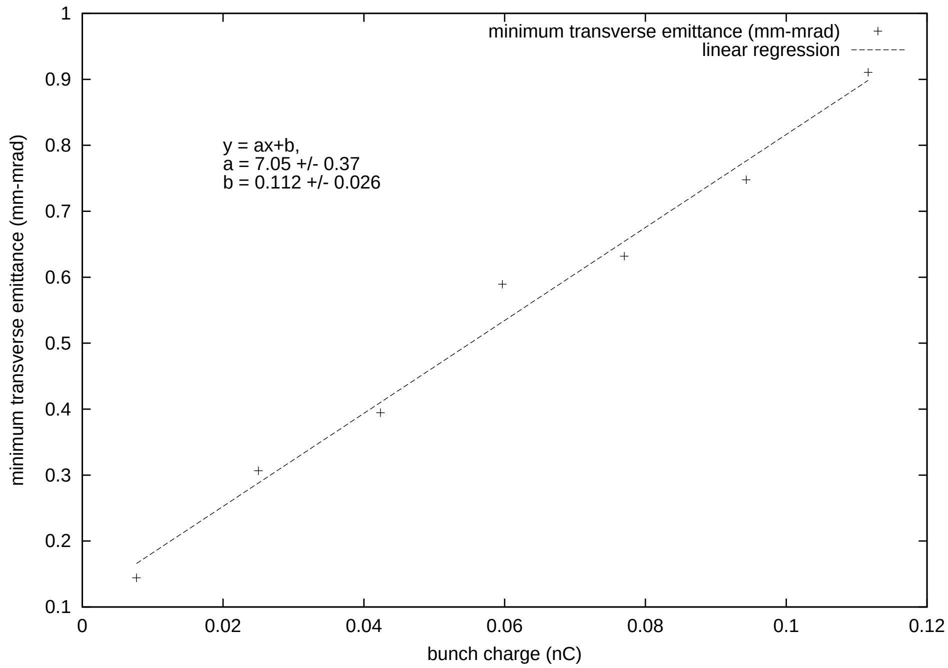

FIG. 9: Plot of minimum possible emittance as a function of charge per bunch.

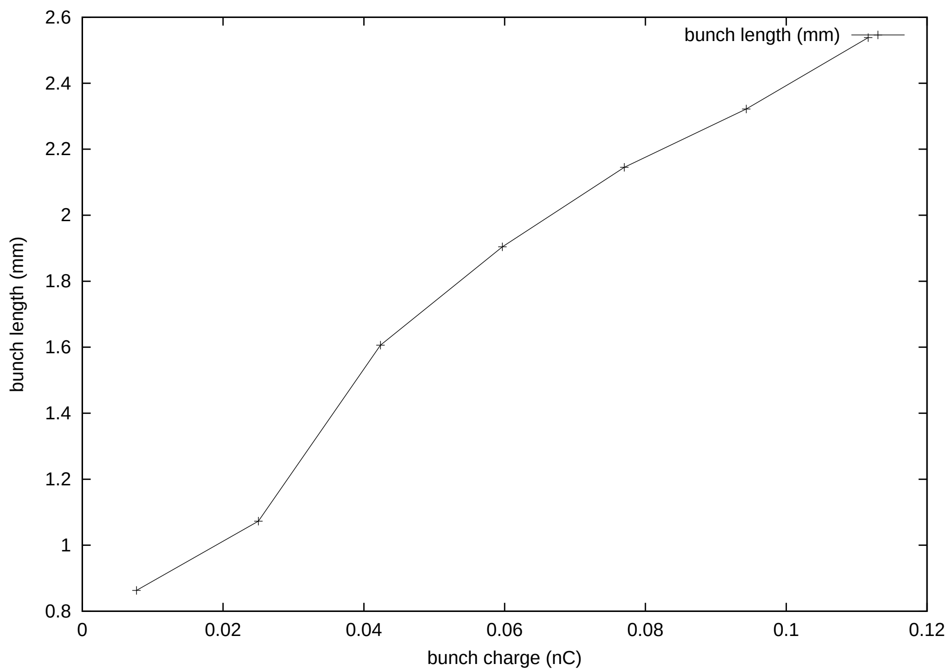

FIG. 10: Plot of bunch lengths corresponding to minimum emittance points from Fig. 9.

ally intensive manner. Having a general idea of where to look for minimal emittance values in the space of parameters we have considered, we can now both refine our search within that space and also begin to consider other possible parameters that have been left relatively untouched, such as gun voltage, gun field configurations, solenoid and cavity positions, independent SRF cavity phase tuning, solenoid field configurations and emittance compensation,

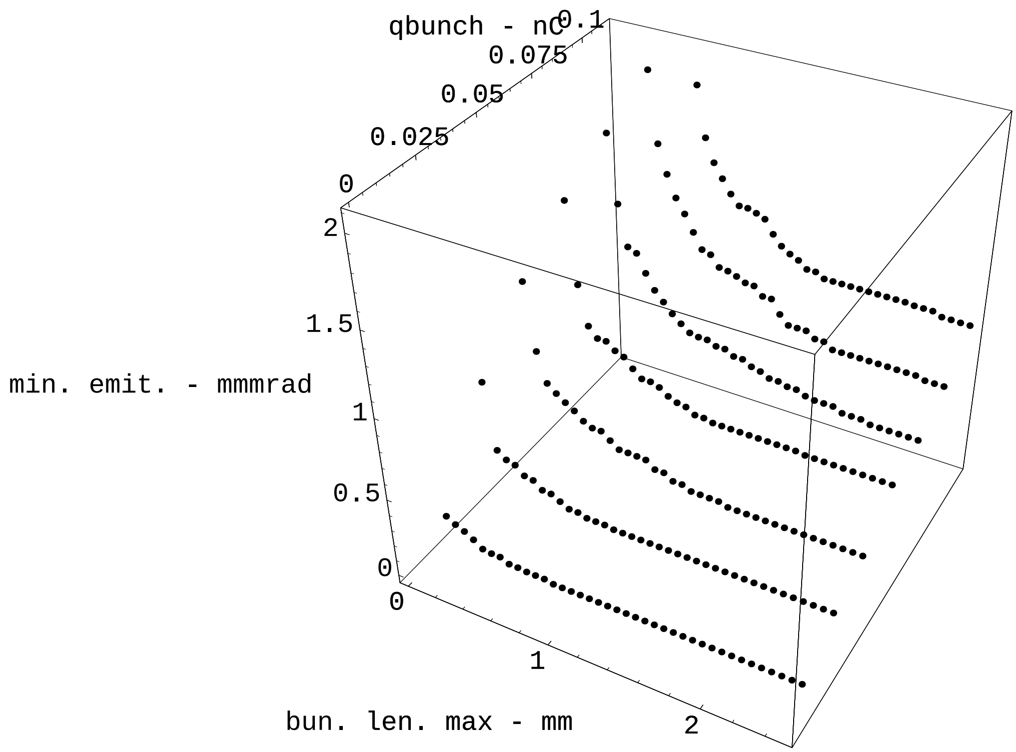

FIG. 11: Plot of minimum possible emittance as a function of bunch charge with an added constraint that the bunch length be less than a specified maximum.

TABLE II: Parameter configurations that minimize emittance for 7.7 pC and 77 pC.

Q (pC)	$\sigma_{r,\rm ini}$ (mm)	B_1 (mT)	B_2 (mT)	$E_{\rm buncher}$ (MV/m)	SRF phases (°)	$\epsilon_{\rm min}$ (μm)	$\sigma_{z,\rm rms}$ (mm)
7.7	0.28	51.25	31.25	1.3	-2	0.14	0.86
77	0.46	57.09	25.63	1.3	-2	0.63	2.14
(77	0.64	54.17	34.06	1.48	-2	0.68	1.72)

etc. In addition, it would be useful to extend the BASH supercomputer framework to jump straight to a minimum-emittance configuration once given a starting point and a region to consider. The simplest form of such an optimizer would use Ivan Bazarov's $3^N - 1$ point box algorithm described in section III, but only consider points on boxes that had minimum values for emittance over the box. The algorithm would finish when a minimum step size has been reached. Once this development has been completed, analysis of the injector using

ASTRA should be simplified greatly, allowing quick and easy access to large amounts of simulation data.

VI. ACKNOWLEDGMENTS

I would like to thank my mentor Dr. Charles Sinclair for his clear, fun instruction and genuine enthusiasm for the project. Also, my thanks go to Dr. Ivan Bazarov for his frequent, interesting lectures on beam dynamics, as well as his nudging and prodding in the right direction when my efforts would begin to run astray. Many thanks also to Prof. Richard Galik for setting a wonderful example that I am sure all of us hope to live up to some day, as well as his effort and dedication towards making the REU program as rewarding as possible, and also to Monica Wesley for making my trip to and stay in Ithaca possible and enjoyable. Last, but not least, I would like to thank Frank Labonte and his supercomputer `Feynman` for their impressive ability to put up with my lack of computer skills, as well as their crucial help with the computing end of my project. This work was supported by the National Science Foundation REU grant PHY-0243687 and research co-operative agreement PHY-9809799.

---

[1] S.M. Gruner, M.Tigner (eds.), CHESS TM 01-003, available at: http://erl.chess.cornell.edu/papers/ERL_Study.pdf

[2] I. Bazarov and C. K. Sinclair, *High Brightness, High Current Injector Design for the Cornell ERL Prototype*, Proceedings of the 2003 Particle Accelerator Conference, Portland, OR; by IEEE, Piscataway, NJ.

[3] A. B. Kowalik, *Designing and Modeling a 750 kV Electron Gun for an Energy Recovery Linac*, Summer 2002; see: http://www.lepp.cornell.edu/public/reu .