As a part of the University of Michigan REU and the CERN Summer Student Programme, I spent nine weeks studying the minimization of beam dimensions in the future linear collider CLIC under the direction of Rogelio Tomas Garcia with the help of Javier Barranco Garcia. I focused on the optimization of the strengths and positions of the nine sextupoles that are a part of the Final Focus System (FFS) in CLIC. The Final Focus System comprises final half-kilometer of the Beam Delivery System (BDS) and is involved in the beam precision at the interaction point.

The virtual manipulation of the beamline elements was performed through the set of MAPCLASS2 codes, which allows the user to operate directly on a beamline or on its transfer map. The beamline is created by loading a Twiss file generated by MAD-X PTC. MAPCLASS2 runs in Python 2.6, and the majority of the work of this summer project was the modification or creation of Python scripts. All MAPCLASS2 functions allow the user to define the order at which calculations will be performed, the default being third order, which allowed for speed in initial calculations.

First studied was the dependence of the beam dimensions, σ_x and σ_y, of the nominal lattice on the order of calculation. This had been researched previously by my advisor Rogelio, and showed that there are significant errors to be corrected below ninth order for both dimensions. Next, several linear searches were performed for adjusted positions and strengths of the nine sextupoles to minimize σ_x only. The optimization of position was always only performed on sextupoles that were surrounded by drifts, which described five of the nine sextupoles. Even considering the adjustment of the two sextupoles closest to the interaction point, isolated adjustment produced very little minimization of σ_x, and I moved on to optimizing multiple strengths or positions simultaneously.

The minimization of σ_x was searched for using Nelder-Mead's Simplex Method, which uses a simplex (shape) with $n + 1$ vertices to search over an n-dimensional region for better and better points. This simplex method does not involve any calculation of gradients and relies on the reflection of vertices across the centroid of the simplex to search for point with smaller function values. The code to implement the simplex method was already written, but the function the used MAPCLASS2 to calculate σ_x needed to be added. When this was finished, I performed a relatively comprehensive study of the minimization of σ_x with the optimization of the strengths of different sextupoles. This study started with adjusting each sextupole strength individually, confirming the ideal strength obtained in the previous linear searches. I tested the optimization of increasing numbers of sextupoles with different initial guesses at different orders to observe the dependence of the rate of convergence, resultant optimized strengths, and minimized σ_x. While some sextupoles regularly converged to similar values with each trial, suggesting a stronger impact on the final beam dimensions, others changed drastically from order to order, from approximately doubling in ideal strength to changing from positive to negative.

A similar procedure was followed for the adjustment of sextupole positions, excepting that an extra step needed to be taken when modifying the position of the magnets. The simplex method has no built-in technique to restrict guesses to a certain range, but the sextupoles could only be moved with a finite space. Initially, any guesses that were outside of the surrounding drifts for a particular sextupole were ignored, but this was not ideal for the minimization method. With no feedback to the simplex method, subsequent guesses would often continue to be out of range instead of closer to original position. This was eventually rectified by halving any position adjustments that were out of range until the values were valid. This guess adjustment was performed with in the 'test-function' script, namely the function that calculated σ_x, and later other values. The modified guesses were then passed back to the simplex script so that further iterations reflected the adjustment.

Once minimized σ_x were found for the modification of all strengths and/or all positions, it was important to determine the impact such changes had on the other beam dimension, σ_y. For the relatively small improvements in σ_x (~ 1%), σ_y increased by two orders of magnitude. In order to minimize both dimensions simultaneously, a weighted χ^2 became the value minimized. This χ^2 was weighted with respect to the 0^{th} order (ideal) values of σ_x and σ_y (40 nm and 1 nm respectively) such that each had an equal weight in χ^2.

Furthermore, in an effort to improve convergence rate and minimization, a few weeks were spent on the implementation of an 'improved' Nelder-Mead Simplex Method, as described by Drs. Pham and Wilamowski of Auburn University in 2011. The method involved the calculation of quasi-gradients with respect to an addiction point comprised of coordinates of the vertices of the simplex. While the modifications to the simplex method were completed, the results were far from ideal. The algorithm could consistently find the minimum of a simple three-dimensional parabolic well, but only very conditionally converged in the search for the minimum of Rosenbrock's parabolic valley, a function that was used to compare the original and improved simplex method in the paper that described the method. A possible explanation for the difference between the published performance and my experience could be the definition of the initial simplex. The code I worked with took a guess for one vertex and change one coordinate at a time to define the remaining *n* vertices. While the simplex method is designed to work for a random simplex, and thus its success of failure should not depend on said simplex, my code by default has vertices with multiple coordinates in common, reducing the 'randomness' of the simplex. The relatively small distance between adjacent vertices meant that the quasi-gradients had a tendency to blow up, and guesses would be far away from the simplex, an inefficient search method. Additionally, as the simplex converged closer to one point, the gradients might become insignificant as function values at adjacent vertices approached each other. When the simplex was not converging on the actual minimum, the small gradients meant the guesses were not far from the existing simplex, and thus not measurably closer to the minimum. Between these two problems, the modified simplex method, which performed worse for higher dimensional simplexes, was completely ineffective for the minimization of the weighted χ^2 by the optimization of the nine strengths and five positions.

In the end, the original simplex method proved sufficient for the minimization of χ^2, though it had a significant dependence on the order to which calculations were performed. The optimized strength and position parameters calculated at third order measured an improvement of χ^2 of about four orders of magnitude, but when the same parameters are used for higher order, more accurate calculations, χ^2 is an order of magnitude worse (larger) than that of the nominal lattice design. With higher order calculations during minimization, χ^2 can be reduced by as much as 99%: for seventh order optimization, χ^2 falls from 2.56×10^{-20} m^2 to 4.90×10^{-22} m^2. This corresponds to a reduction in σ_x and σ_y from 40.2 nm to 40.1 nm and 1.16 nm to 1.02 nm, respectively, a 1.7% reduction in beam area.

This work has much potential for continued use and improvement. This simplex method for minimization can be expanded to take into account other elements of the beamlines, such as the quadrupole magnets, or other aspects of the sextupoles, such as tilts. However, with increased dimensionality of the parameter space, it becomes more important to properly and effectively implement an improved simplex method that will reduce the time/iterations required to converge. The aforementioned modifications to the simplex method are not the only suggested ones, but warrant further investigation into their use. The set of scripts written can also be more fine-tuned by increasing the limits on the regions in which optimized strengths and positions are searched for. Now that the script has been written for the definition of the range of position, it will be easy to reapply elsewhere. These methods are applicable to other accelerators besides CLIC and become a valuable tool at CERN.

A summary of my complete results and scripts will be made available to my advisor and relevant files will be appended to MAPCLASS2.

References:

Tomás, R. "Nonlinear Optimization of Beam Lines." *Physical Review Special Topics - Accelerators and Beams* 9.8 (2006): n. pag. Print.

J. A. Nelder, R. Mead, “A Simplex Method for Function Minimization”, *Computer Journal*, vol. 7, pp. 308-313, 1965

N. Pham, B. M. Wilamowski, “Improved Nelder Mead’s Simplex Method and Applications”, *Journal of Computing*, vol. 3, iss. 3, pp. 55-62