GENETIC OPTIMIZATIONS OF RF CELL PROFILE USING SUPERFISH[*]

M. Kaemingk[†], P. M. Anisimov, L. Duffy, J. W. Lewellen, K. Shipman, B. Sims, H. Xu
Los Alamos National Laboratory, Los Alamos, NM, United States

Abstract

Important radio-frequency (RF) cell parameters such as the shunt impedance R_s, the quality factor Q, and the ratio of accelerating to peak field, depend on the cell geometry. Thus, it is desirable to optimize the cell profile to maximize the performance of the cell. The use of B-splines to optimize cell profiles with an evolutionary algorithm has previously been demonstrated. The RF field solver Superfish, however, does not support defining cell geometries using B-splines. Here we discuss a similar approach using only linear and elliptical segments supported by Superfish to define the cell geometry, making it possible to carry out genetic optimizations with Superfish.

INTRODUCTION

The effectiveness of radio-frequency (RF) cells in accelerating and bunching particle beams depends on parameters such as the shunt impedance R_s, the unloaded quality factor Q, and the maximum achievable accelerating gradient, which is limited by the peak field the cell can sustain. These parameters are in turn dependent on the cell geometry. Optimizing the cell profile for any particular application is therefore necessary to maximize the cell performance. Evolutionary optimizations have been used for this purpose, as described in Ref. [1]. Evolutionary optimizations require performing large numbers of simulations, making light-weight RF field solver codes like Superfish [2] well suited to carry them out.

In the work described in Ref. [1], the cell profile was defined as a B-spline. The positions of the points defining the B-splines could then be varied to optimize the cell profile. Superfish does not support the use of B-splines. In Superfish, the cell profile can be defined by a series of elliptical and linear segments. The parameters defining these segments can be varied to perform the optimization.

In this work, we describe a system developed for performing genetic optimizations of cell profiles using Superfish. This system has been implemented in Python which allows the use of packages such as PySuperfish [3] and Xopt [4]. These packages and the flexibility offered by Python enable the use of a variety of optimization approaches beyond genetic algorithms, and make it easy to optimize for many different possible objectives. In the last section we show a basic example of a genetic optimization performed using this system.

SYSTEM DESCRIPTION

In this section we describe the optimization approach. In general, the workflow can be described as an iterative procedure consisting of the following steps:

1. Generate parameters for defining the cell profile
2. Validate the cell profile
3. Perform simulations with Superfish
4. Evaluate results

Each of these steps is described in greater detail in the following subsections.

Cell Profile Parameters

As discussed in the introduction, we define the cell profile using a series of elliptical and linear segments. Each segment is defined by the segment endpoints as well as the slopes of the tangent lines at each endpoint. Defining the segments in this way makes it easy to ensure that the segments define a continuous curve with a continuous derivative. Therefore, the optimization variables are the endpoint coordinates of the segments and the endpoint tangent slopes. The number of segments is determined from the number of endpoints and endpoint slopes given.

Superfish only allows segments from axis-aligned ellipses, which can be described by the equation $\frac{(z-h)^2}{a^2} + \frac{(r-k)^2}{b^2} = 1$. This can be rewritten as

$$(z-h)^2 + c(r-k)^2 = a^2. \tag{1}$$

This can be differentiated to obtain expressions for the tangent slopes. Substituting in endpoint coordinates (z_n, r_n) and (z_{n+1}, r_{n+1}) and endpoint slopes s_n, s_{n+1}, conveniently yields a set of linear equations which can be expressed in matrix form as follows

$$\begin{pmatrix} h \\ ck \\ c \end{pmatrix} = M \begin{pmatrix} z_n^2 - z_{n+1}^2 \\ z_n \\ z_{n+1} \end{pmatrix}, \tag{2}$$

where

$$M = \begin{pmatrix} 2(z_n - z_{n+1}) & 2(r_n - r_{n+1}) & -(r_n^2 - r_{n+1}^2) \\ 1 & s_n & -s_n r_n \\ 1 & s_{n+1} & -s_{n+1} r_{n+1} \end{pmatrix}.$$

Together with Eq. (1) and $c = a^2/b^2$, Eq. (2) can then be used to find h, k, a, b, which are the parameters used by Superfish to define ellipse segments. Thus the optimizer (e.g. Xopt) generates a set of coordinates $(z_1, r_1), (z_2, r_2), ..., (z_{N+1}, r_{N+1})$ and slopes $s_1, s_2, ..., s_{N+1}$. The ellipse parameters are then obtained using Eqs. (1) and (2). Figure 1 shows an example of a cell profile defined in this way.

---

[*] Work supported by U. S. Department of Energy through the Laboratory Directed Research and Development program of Los Alamos National Laboratory, under project number 20260114ER.

[†] kaemingk@lanl.gov

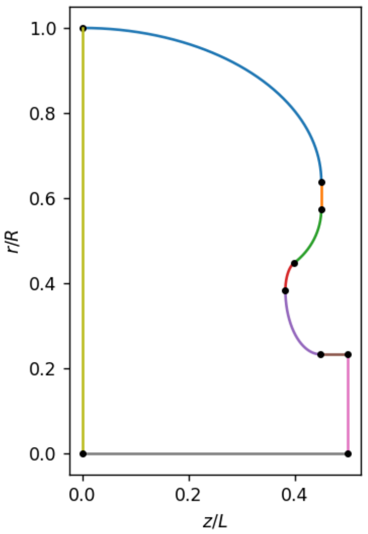

Figure 1: Example of profile generated using elliptical segments for Superfish simulations. This example is the profile resulting from the optimization discussed in the subsequent section.

Profile Validation

The procedure described can result in invalid parameters or unphysical profiles. If $c < 0$, the parameters h, k, a, b define a hyperbola instead of an ellipse. Furthermore, as seen in Fig. 2, the resulting segments can intersect or meet at a cusp. Cusps can result because two ellipse segments, a long one and a short one, are both defined by the same set of endpoints and slopes. Superfish always uses the short segment, but the segment that meets the neighboring segment to form a smooth joint may instead be the long segment. In this case, two neighboring short segments meet forming a cusp. This can easily be identified by computing the tangent vectors of neighboring segments. If the scalar product between these vectors is negative, the two segments meet at a cusp, and the profile is invalid. In summary, after the segment parameters have been obtained, we check if $c < 0$ (this can be done analytically because the segments are linear and elliptical), if there are intersections, and if any segments meet at a cusp. If any of these is true, the profile is discarded.

Superfish Simulations

Once a valid profile has been generated, an autofish (.AF) file for Superfish is created. This file requires a header setting parameters such as mesh resolution, boundary conditions, material properties, etc., followed by lines with the segment parameters defining the cell geometry. The header is copied from a file indicated by the user. The use of a header file allows flexibility to define the Superfish problem as desired. An autofish file is created for each cell profile which is then used by PySuperfish to perform the RF simulation.

PySuperfish can be used to run Superfish on containers, allowing it to be run on Linux. This option offers a great advantage as most computer clusters run Linux operating systems, as is the case for the high-performance computing cluster at Los Alamos National Laboratory, used for the work presented here. PySuperfish can also be used to load the simulation results. Figure 3 shows the RF fields computed from simulating the profile shown in Fig. 1 using PySuperfish.

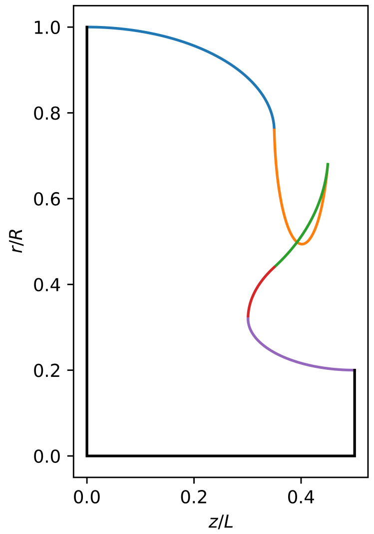

Figure 2: Example of invalid profile.

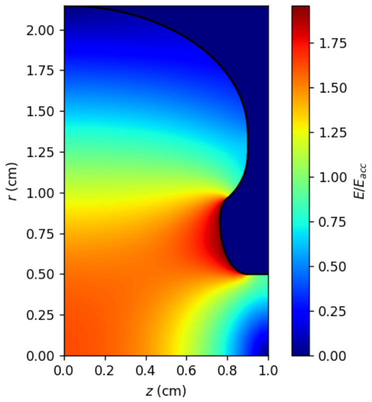

Figure 3: Electric fields from Superfish simulations of profile shown in Fig. 1.

Results Evaluation

Once the simulations are completed, the output is analyzed to obtain the values of the objective variables. Xopt uses these to generate a new set of parameters. To allow flexibility in choosing the objectives, the analysis function is defined separately and specified by the user. The function takes a PySuperfish object used to load the simulation results which are used to compute the desired objectives and other desired quantities which are then returned in dictionary form. Multiple objectives and constraints can be set.

In summary, Xopt generates a set of endpoint coordinates and slopes which are used to find the parameters for linear and elliptical segments to define a cell profile. The profiles are then validated and written into autofish files. PySuperfish is used to simulate the RF fields and cell parameters. The desired quantities including the optimization objectives are computed, and the process is repeated.

EXAMPLE OPTIMIZATION

Now we show an example optimization performed using the system described. The objective was

$$G = \frac{R_\mathrm{s}}{E_\mathrm{peak}/E_\mathrm{acc}},$$

as also used in Ref. [1]. E_peak is the peak surface electric field and R_s is the time-dependent shunt impedance per unit length given by $E_\mathrm{acc}^2 L/P_\mathrm{diss}$, where $L = 1\,\mathrm{cm}$ is the cell length and P_diss is the dissipated power. The accelerating gradient is given by $E_\mathrm{acc} = \frac{1}{L}\int_{-L/2}^{L/2} E_z(0,z)\cos(\omega z/c)dz$. Table 1 provides the ranges over which the parameters are varied. This optimization is a proof of principle and not for any particular application. The coordinates and slopes for the endpoints are given by

$$z_i = u_i Z,\ r_i = v_i(R-r)+r,\ s_i = t_i R/Z.$$

See Fig. 4 for reference on the values and definitions of R, Z, and r.

Table 1: Optimization parameters and ranges for example optimization described.

Point	***u***	***v***	***t***
1	0	1	0
2	1	0.3 – 0.9	-∞
3	1	0.3 – 0.8	-∞
4	0.7 – 1	0.1 – 0.5	0 – 5
5	0.5 – 0.9	0.05 – 0.3	∞
6	0.6 – 1.1	0	0
7	L/Z	0	0

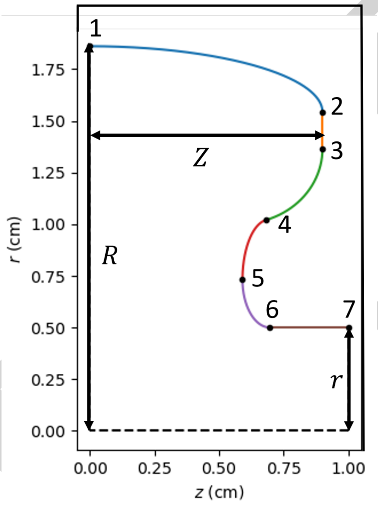

Figure 4: Reference figure for parameters used in the example optimization. $Z = 0.9\,\mathrm{cm}$, and R could vary between 1.7 cm and 2.3 cm with the frequency constrained to 5.712 ± 0.1 GHz

Figure 5 shows the convergence of the objective value G as the optimization progresses. The profile and corresponding fields shown in Fig. 1 and Fig. 3 are the final optimized cell profile and fields. The time-dependent shunt impedance per unit length for this cell is 85.7 MΩ/m with a peak surface electric field to accelerating gradient ratio of 1.96. These values assume a beam traveling at the speed of light.

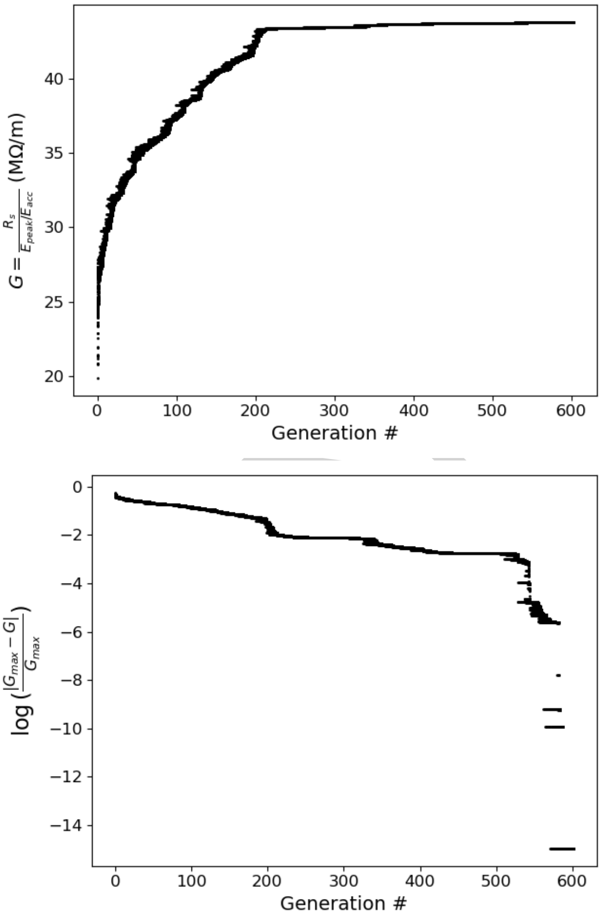

Figure 5: (Top) Value of optimization objective for each generation. (Bottom) Semi-log plot of the fractional error of the objective relative to the final objective value. The plot scale is limited by machine precision.

CONCLUSION

We have developed a system for performing RF cell profile optimizations using Superfish. This system is well suited for genetic optimizations similar to those described in Ref. [1]. The optimization parameters consist of a series of points and slope values that define a profile consisting of linear and elliptical segments forming a smooth curve (i. e. continuously differentiable). This system leverages Xopt and PySuperfish to enable a variety of optimization approaches and parallelization and makes it easy to analyze the simulation results and define optimization objectives. As a proof of principle, we show the results of an optimization of a cell optimized to maximize the ratio of the time-dependent shunt impedance to the peak surface field normalized by the accelerating gradient.

REFERENCES

[1] M. H. Nasr and S. G. Tantawi, “New geometrical-optimization approach using splines for enhanced accelerator cavities’ performance”, in *Proc. IPAC’18*, Vancouver, Canada, Apr.–May 2018, pp. 4395–4397. doi:10.18429/JACoW-IPAC2018-THPMK049

[2] M. T. Menzel and H. K. Stokes, “User’s guide for the POISSON/SUPERFISH group of codes”, Los Alamos National Laboratory, Los Alamos, NM, USA, Rep. LA-UR-87-115, Jan. 1987.

[3] C. Mayes, “Pysuperfish: python tools for poisson superfish calculations”, `https://github.com/ChristopherMayes/PySuperfish`, Version 0.1.2; accessed 2026-04-30, 2024,

[4] R. Roussel, C. Mayes, A. Edelen, and A. Bartnik, “Xopt: A simplified framework for optimization of accelerator problems using advanced algorithms”, in *Proc. IPAC’23*, Venice, Italy, pp. 4847–4850, Sep. 2023. doi:10.18429/JACoW-IPAC2023-THPL164