Beam Physics Note
TRI-BN-22-26
August 24, 2022

Updated Multipole Tuning Algorithm

Josh Kraan

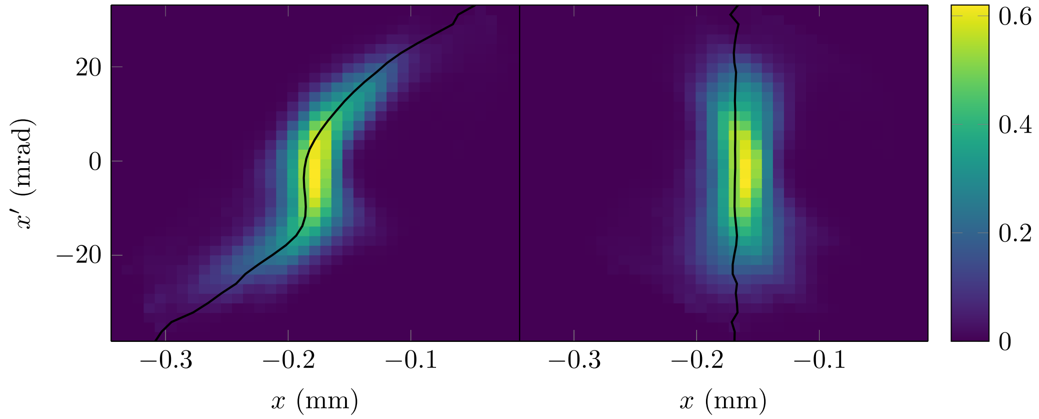
TRIUMF

Abstract: A previously developed multipole tuning algorithm was specific to one beam energy and often provided solutions with excessively high voltages. The updated tuning algorithm presented in this beam physics note uses a model for the multipole shown in TRI-BN-22-22 and regularization methods to allow it to function for any beam energy and provide lower-voltage solutions, with a significantly simpler implementation.

1 Introduction

An algorithm to tune the multipole corrector of the TRIUMF Canadian Rare isotope facility with Electron Beam ion source (CANREB) High Resolution Separator (HRS) based on exit emittance scans was previously developed by Dan Sehayek [1]. In TRI-BN-22-22 [2] a new model that could be used to replace ray-tracing code in this algorithm was detailed, which would allow the algorithm to function for any beam energy. In this beam physics note, an updated algorithm that uses the new model along with regularization of the least-squares problem to provide better results with a simpler codebase is shown.

The horizontal emittance scanner at the exit of the HRS measures beam intensity as a function of horizontal position and horizontal angle. To maximize resolving power, it is desired that the beam have as small positional spread at the exit as possible. Due to high-order aberrations, the beam at the HRS exit may become deformed, such as is shown in Figure 1a. The HRS multipole is intended to correct these aberrations and produce a beam with minimal positional spread, as shown in Figure 1b.



(a) Original

(b) Corrected

Figure 1: Example emittance scans, showing what a beam may look like before and after correction. Note that this data was not collected on the HRS, and the corrected data is a calculated prediction. The centroidal position is graphed for each as a solid black line.

2 Tuning Algorithm

At the HRS exit, the multipole mainly causes a constant amount of displacement in position, the amount of which varies as a function of angle. To determine the displacement required to minimize the positional spread, the centroidal position as a function of angle is calculated as:

$$\bar{x}(x') = \frac{\sum_{i=0}^{n} x_i I(x_i, x')}{\sum_{i=0}^{n} I(x_i, x')} \tag{1}$$

The centroidal position of the beam is shown in Figure 1. Ideally this would be constant over angle.

In a previous beam physics note [2] a method for calculating the displacement as a function of angle at the HRS exit for each pole pair was shown, and the same method will be used here. Figure 2 shows the displacement for pole pairs 11 to 13 at 1 V, as well as the desired displacement for the example emittance scan shown in Figure 1a. The overall displacement of the multipole can be found by multiplying the displacement of each pole by its voltage and adding all the displacement functions together. Clearly these displacement functions are non-orthogonal, which means that the pole voltages required for a given displacement cannot be calculated using a method similar to Fourier decomposition. Instead, the method of least-squares can be used.

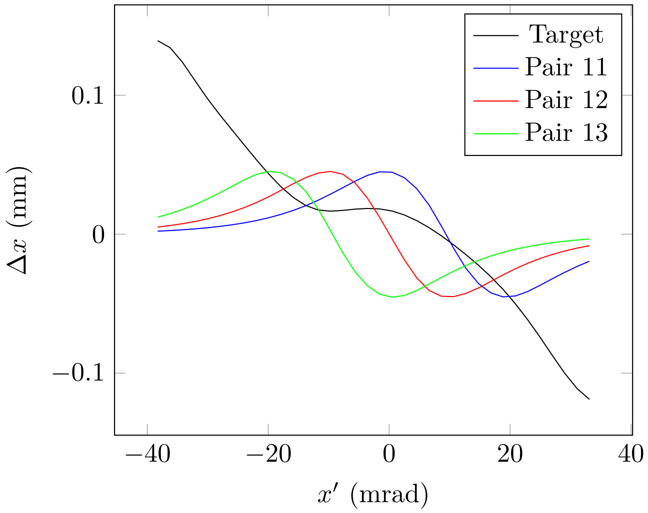


Figure 2: Displacement functions for pole pairs 11 to 13 graphed with the target displacement necessary to correct the aberrations in Figure 1a.

If angle is discretized into m sections and the pole pair voltages are described by a vector $\boldsymbol{w}$ with $n = 23$ entries, then the displacement as a function of angle for multiple poles operating at the same time would be $\boldsymbol{Mw}$, where $\boldsymbol{M}$ is the $m \times n$ displacement matrix, defined as:

$$\boldsymbol{M} = \boldsymbol{\Delta_1 e_1^T} + \boldsymbol{\Delta_2 e_2^T} + \cdots + \boldsymbol{\Delta_n e_n^T}, \tag{2}$$

where $\boldsymbol{\Delta_i}$ is the displacement as a function of angle for pole pair i at 1 V, and $\boldsymbol{e_i}$ is a unit vector with a 1 in the ith place.

The least-squares problem is then:

$$\boldsymbol{Mw} \cong b - \bar{\boldsymbol{x}} \tag{3}$$

Where b is the corrected beam position. This can either be considered an additional free parameter and solved for as an intercept, or it can be fixed to a desired position, such as the position with the maximum intensity. Allowing b to vary can result in better solutions when regularized, so both methods were implemented.

In this case, directly computing the least-squares solution results in a poor solution with excessively high voltages on the edge poles. The cause of this can be seen by using singular value decomposition.

2.1 Singular Value Decomposition

The material shown here comes from *Scientific computing: an introductory survey* [3].

The process of singular value decomposition decomposes the displacement matrix into the form:

$$\boldsymbol{M} = \boldsymbol{U\Sigma V}^T, \tag{4}$$

where all entries of $\boldsymbol{\Sigma}$ are zero except for the diagonals, which are sorted in non-ascending order ($\sigma_{i-1} \geq \sigma_i$) and are called the singular values of $\boldsymbol{M}$. The displacement matrix in this analysis is full-ranked, so the degenerate case will not be considered here. The columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are sets of mutually orthogonal unit vectors, known as the left and right singular vectors respectively.

Denoting $\boldsymbol{u_i}$ and $\boldsymbol{v_i}$ as the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, this can be expanded to:

$$\boldsymbol{M} = \sigma_1 \boldsymbol{u_1} \boldsymbol{v_1}^T + \sigma_2 \boldsymbol{u_2} \boldsymbol{v_2}^T + \cdots + \sigma_n \boldsymbol{u_n} \boldsymbol{v_n}^T \tag{5}$$

Comparing Equation (5) with Equation (2), it can be seen that the singular value decomposition has changed the 23 original displacement functions, each caused by a single pole pair set to 1 V, into 23 orthogonal displacement functions $\boldsymbol{u_i}$ corresponding to a vector of pole pair voltages $\boldsymbol{v_i}$. The singular value σ_i corresponding to each orthogonal function describes how much influence that function has on the displacement. For example, if $\sigma_1 \gg \sigma_n$, then $\boldsymbol{Mv_1} \approx \boldsymbol{M}(\boldsymbol{v_1} + \boldsymbol{v_n})$, so significantly different solutions for pole voltages could result in essentially the same displacement. The ratio between the highest singular value and lowest of a matrix is known as the condition number, $\text{cond}_2(\boldsymbol{M}) = \sigma_1/\sigma_n$.

Figure 3 shows the singular values of the displacement matrix divided by the first singular value. It can be visually seen that the condition number is on the order of 10^4, and even if the two lowest terms were dropped the condition number would be on the order of 10^3.

The orthogonal displacement functions and the pole voltages corresponding to the first and last singular values are graphed in Figures 4a and 5a, showing the configurations with the largest and smallest impact on displacement respectively. In Figure 3 it can be seen that the 2nd and 3rd lowest singular values are very close in value; the corresponding configurations are shown in Figures 4b and 5b, showing that both cause a ripple with the same frequency and amplitude, the only different being a 180° phase shift halfway though the angle range.

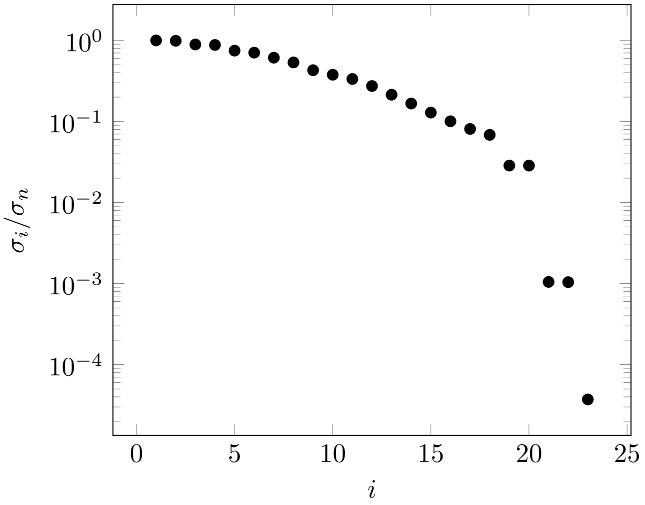
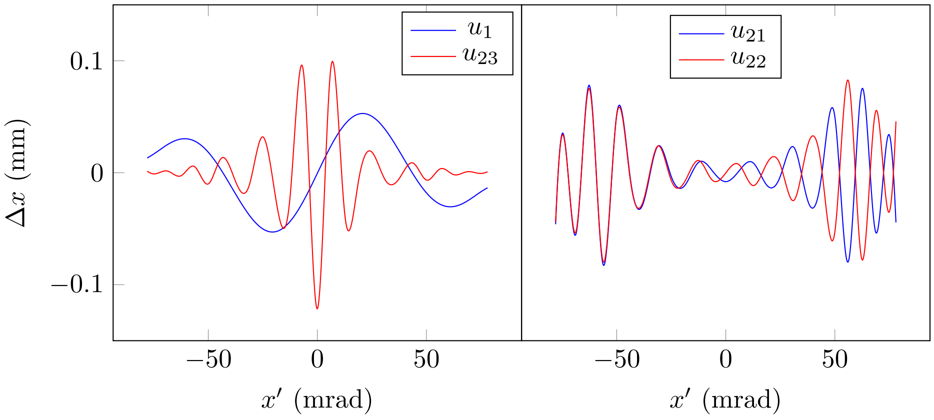


Figure 3: Singular values for the displacement matrix, normalized by dividing by the first singular value.



(a) u_1 & u_{23}

(b) u_{21} & u_{22}

Figure 4: Orthogonal displacement functions produced by singular value decomposition of the displacement matrix.

For a constant b, the least-squares solution can be rewritten in terms of the singular value decomposition:

$$\boldsymbol{w} = \sum_{\sigma_i \neq 0} \frac{\boldsymbol{u_i} \cdot (b - \bar{\boldsymbol{x}})}{\sigma_i} \boldsymbol{v_i} \tag{6}$$

For large condition numbers, small changes in $b - \bar{\boldsymbol{x}}$ can result in large changes in the least-squares solution $\boldsymbol{w}$. One solution to this problem is to truncate the sum in Equation (6), ignoring displacement functions with singular values below some threshold. Another solution is to take advantage of the freedom to add vectors of pole pairs voltages corresponding to these low singular values in order to obtain a "nicer" solution for pole pair voltages. The latter approach will be used here.

2.2 Regularization Methods

Regularization introduces some bias in order to reduce the variance of least-squares solutions for ill-conditioned problems. Different biasing methods can be chosen based on the desired solution behavior. Typically, biasing is done based on a norm of the independent variable, such as the L_0 pseudo-norm (number of non-zero elements), the L_1 norm (sum of absolute values of elements), or the L_2 norm (Euclidean norm).

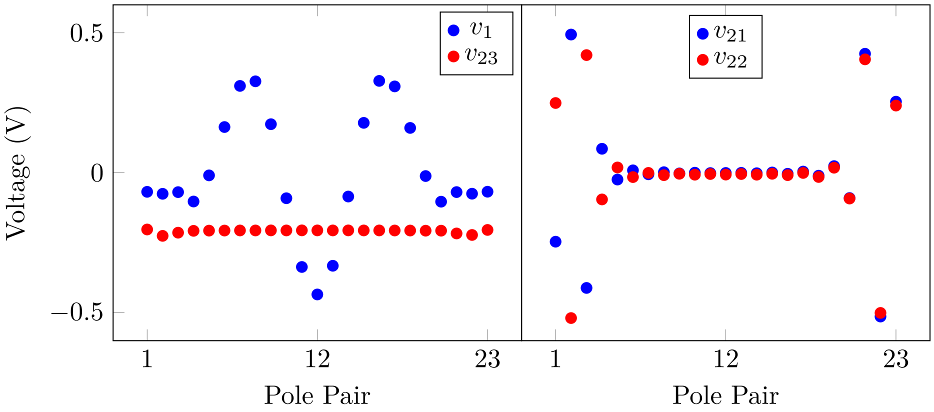
Scikit-learn [4] includes several efficient methods for least-squares with regularization. A brief summary of relevant methods, based on the Scikit-learn documentation, is included here for completeness. To match the documentation the least-squares problem considered in the following subsections will be $\boldsymbol{X}\boldsymbol{w} \cong \boldsymbol{y}$.

2.2.1 Ridge Regression

Ridge regression is based on the L_2 norm, and minimizes the objective function

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \alpha \|\boldsymbol{w}\|_2^2, \tag{7}$$

where α is a constant that controls the regularization strength. Unlike all the other methods considered here, the solution can be directly calculated with standard least-squares methods.



(a) v_1 & v_{23}

(b) v_{21} & v_{22}

Figure 5: Pole pair voltages corresponding to the displacement functions of Figure 4. Note that v_{23} is nearly constant, implying that adding a constant voltage to all pole pairs will not significantly affect the multipole function.

2.2.2 Lasso Regression

Lasso regression is based on the L_1 norm, and minimizes the objective function

$$\frac{1}{2n_{samples}}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2+\alpha\|\boldsymbol{w}\|_1. \tag{8}$$

The solution to Lasso regression cannot be directly calculated, and must be found through iteration. Due to the use of the L_1 norm, it tends to prefer solutions with fewer non-zero coefficients.

2.2.3 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit regularizes on the L_0 pseudo-norm of the independent variable, $\|\boldsymbol{w}\|_0$. It can be used to search for the least-squares solution for a given number of non-zero coefficients, or the solution with the minimum number of non-zero coefficients for a given tolerance on the least-squares solution.

2.2.4 Elastic Net

The Elastic Net method uses both the L_1 and L_2 norms. It minimizes the objective function

$$\frac{1}{2n_{samples}}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2+\alpha\rho\|\boldsymbol{w}\|_1+\frac{\alpha(1-\rho)}{2}\|\boldsymbol{w}\|_2^2. \tag{9}$$

Here again α controls the regularization strength, and an additional constant ρ that controls the relative strengths of the L_1 and L_2 penalties has been added. For $\rho=0$ this reduces to Ridge regression, and for $\rho=1$ to Lasso.

In this context of this paper many of Elastic Net's advantages, such as its behavior when predictors are highly correlated, are not very relevant. However, an advantage is that Elastic Net can prefer sparse solutions like Lasso while also applying a quadratic penalty on the independent variable like Ridge.

3 Performance Estimates

All the previously described methods were tested on a Zgoubi [5] simulation of the HRS, using the `30kV_U1+_multiparticle.in` file created by Owen Lailey [6]. The aberrations in this model are minor, with the mean absolute deviation of the centroids being 15.4 µm, so little correction was required. The emittance data before correction is shown in Figure 6.

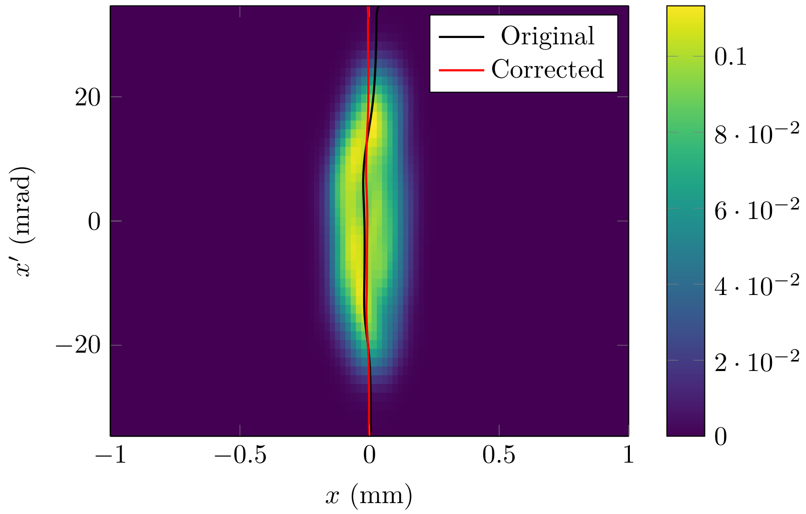


Figure 6: Emittance data from the Zgoubi simulation of the HRS before correction. The centroidal position is graphed before and after correction with the Lasso scheme.

The pole pair voltages output by each model are shown in Table 1, and the mean absolute deviation of the corrected beam for each model is shown in Table 2. All methods performed well in correcting the beam; the difference between the worst and best mean absolute deviation was less than 0.5 µm. Interestingly, despite theoretically having the lowest deviation, the unconstrained "DSehayek" method performed the worst.

	Calculated Voltages (V)					
Pole Pair	DSehayek*	OLailey*	Ridge	Lasso	OMP	Elastic
1	−1000.000	0.544	−0.021	0	0	0
2	−551.812	0.680	−0.025	0	0	0
3	1000.000	0.841	−0.032	0	0	0
4	−11.838	2.387	−0.099	0	0	0
5	8.324	6.441	−0.244	0	0	0
6	5.091	15.896	−0.523	0	0	−0.224
7	−1.854	−13.193	−0.491	−0.937	−1.328	−0.761
8	0.378	2.461	0.329	0	0	0.013
9	−0.296	−1.007	−0.167	−0.213	−0.400	−0.186
10	0.325	0.407	0.402	0.182	0.059	0.208
11	0.033	−0.221	0.277	0.121	0.011	0.157
12	−0.261	−0.257	−0.100	−0.236	−0.422	−0.221
13	0.094	−0.043	0.344	0.124	0.068	0.165
14	−0.632	−0.465	−0.285	−0.367	−0.606	−0.353
15	−0.028	−0.235	0.293	0	0.022	0.043
16	−0.745	0.522	−0.088	0	−0.378	0
17	−0.663	−0.938	0.664	0	0	0.081
18	1.170	−0.514	−0.157	0	0	0
19	−260.063	−0.224	−0.115	0	0	0
20	−135.775	−0.087	−0.049	0	0	0
21	1000.000	−0.026	−0.016	0	0	0
22	1000.000	−0.021	−0.013	0	0	0
23	999.936	−0.017	−0.010	0	0	0

Table 1: Pole pair voltages calculated for the input shown in Figure 6, using several regularization methods. The "DSehayek*" method is the original algorithm written by Dan Sehayek [1], and "OLailey*" is the same algorithm with $NoS = 3$, $start = 1$ as suggested by Owen Lailey [6]. For all regularization methods the beam position was fixed. Hyperparameters were chosen based on judgement; it was found that $\alpha = 1 \times 10^{-6}$ provided good results for Ridge, Lasso, and Elastic Net, $\rho = 0.5$ worked well for Elastic Net, and a tolerance of 1×10^{-4} worked for Orthogonal Matching Pursuit.

Method	Mean Absolute Deviation (µm)	Number of Active Pole Pairs	Maximum Absolute Voltage (V)
DSehayek*	3.23	23	1000.0
OLailey*	2.79	23	15.9
Ridge	2.77	23	0.7
Lasso	3.17	7	0.9
OMP	2.83	9	1.3
Elastic	3.12	11	0.8

Table 2: Summary of the configurations specified by each regularization method. The mean absolute deviation of the corrected beam centroids was found using a Zgoubi simulation. Before correction the mean absolute deviation was 15.4 µm. For method hyperparameters and an explanation on the starred methods see the caption of Table 1.

All the regularization methods tested were able to lower the required voltages, and those based on the L_1 or L_0 norms were able to significantly reduce the number of required active pole pairs. The Lasso method is recommended for future work, as it often provides a simple well-performing solution. Orthogonal Matching Pursuit often performs well, but has occasionally failed to converge for specific emittance data, so it is not recommended. Elastic Net may be a good compromise if the voltages recommended by Lasso are too large.

4 Recommendations for Future Work

The following are recommended:

- The current high-level application at `https://devel.hla.triumf.ca/atom` uses the old algorithm, based on a Zgoubi simulation that contains errors [6]. This should be rewritten with the new MultipoleOptimizer class, and options for the beam energy and regularization parameters should be added.
- Both the new and old algorithms require that emittance data be first processed with Rick Baartman's Matlab scripts located at `http://lin12.triumf.ca/emittance/emitmat/`. This processing could be implemented into the MultipoleOptimizer class.
- Currently, all angles with non-zero intensity are weighted equally in the least squares problem. Weighting based on the intensity could be introduced.
- The method shown here recommends the solution with the smallest possible pole pair voltages. If the multipole performs poorly for small voltages, a multiple of v_{23} could be added to the solution.

5 Conclusion

The updated multipole tuning algorithm performs well on Zgoubi simulations, with the Lasso regularization method providing the nicest solutions. Given the equivalent simulated performance with nicer solutions for pole pair voltages, it is recommended that the previous algorithm be replaced with this one for upcoming tests of multipole tuning.

A The MultipoleOptimizer Class

A new MultipoleOptimizer class was written that replaces the files `Optimize.py`, `TableData.py`, `ZPSModuleCentralizer.py`, `ZPSModuleCentroid.py`, and `ZPSModuleDataHandler.py`. Through increased use of vectorized operations, the use of Scikit-learn, and the new method for calculating pole displacement [2], the MultipoleOptimizer class provides improved functionality despite the code being one third the length of the original. Example usage is shown in Listing 1.

```python
import pathlib
from multipole_optimizer import MultipoleOptimizer

data_directory = pathlib.Path('emittance_data')

optimizer = MultipoleOptimizer(charge=1.602176487E-19,
                               energy=30 * 1.6022e-16,
                               free_position=False)
optimizer.method = 'lasso'
optimizer.alpha = 1e-6

# Note that emittance data should be pre-processed to remove noise
```

```
optimizer.optimize_multipole(data_directory / 'TOMO_FIT.txt')

voltages = optimizer.voltages
```

Listing 1: Example usage of the MultipoleOptimizer class. For parameter documentation see `multipole_optimizer.py`. For more detailed examples, see `multipole_optimizer.ipynb`.

B Source Code

All relevant code was archived at a common location on Gitlab.

- The implementation of the MultipoleOptimizer class is located at `https://gitlab.triumf.ca/hla/atom/-/blob/master/multipole_code_and_docs/multipole_jkraan/multipole_optimizer.py`. A Jupyter notebook of the same name shows example usage.
- The Zgoubi simulation used in this report is located at `https://gitlab.triumf.ca/hla/atom/-/blob/master/multipole_code_and_docs/multipole_olailey/Multipole.py`.
- The original optimization code is located at `https://gitlab.triumf.ca/hla/atom/-/tree/master/multipole_code_and_docs/multipole_olailey/hla_code_multipole`.

References

[1] D. Sehayek, R. Baartman, C. Barquest, J. Maloney, M. Marchetto, and T. Planche, "Multipole Tuning Algorithm for the CANREB HRS at TRIUMF," *Proceedings of the 9th Int. Particle Accelerator Conf.*, vol. IPAC2018, 2018, Artwork Size: 3 pages, 1.650 MB ISBN: 9783954501847 Medium: PDF Publisher: JACoW Publishing, Geneva, Switzerland. DOI: `10.18429/JACOW-IPAC2018-THPML079`. [Online]. Available: `http://jacow.org/ipac2018/doi/JACoW-IPAC2018-THPML079.html`.

[2] J. Kraan, "TRI-BN-22-22 - Simple Method for Calculating HRS Multipole Effects Directly from Field Maps," en, p. 8, Jul. 2022.

[3] M. T. Heath, *Scientific computing: an introductory survey* (Classics in applied mathematics 80), Revised second edition, SIAM edition. Philadelphia: Society for Industrial and Applied Mathematics, 2018, ISBN: 978-1-61197-557-4.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] F. Méot, "The ray-tracing code Zgoubi," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 427, no. 1, pp. 353–356, 1999, ISSN: 0168-9002. DOI: `https://doi.org/10.1016/S0168-9002(98)01508-3`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0168900298015083`.

[6] O. Lailey, "CANREB High-Resolution Separator Commissioning Progress," en, Aug. 2021.