ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-WEP029

DEVELOPMENTS IN LUME-ACE3P INCLUDING S-PARAMETER OPTIMIZATION FOR S3P

L. Fowler, D. Bizzozero, SLAC National Accelerator Laboratory, Menlo Park, CA, USA

Abstract

We present here the introduction of optimization to LUME-ACE3P (LUME: Lightsource Unified Modeling Environment; ACE3P: Advanced Computational Electromagnetics 3D Parallel). LUME-ACE3P is a Python wrapper that streamlines workflows for ACE3P, a suite of finite element solvers for electromagnetic fields in complex geometries. LUME-ACE3P offers parameter sweep capabilities, which was previously the only means to perform optimization with this code. In the integration of LUME-ACE3P with the optimization package Xopt, we facilitate efficient and easy to use optimization for accelerator component design. We present the LUME-ACE3P-Xopt workflow with an example problem.

COMPONENT CODES

ACE3P

ACE3P (Advanced Computational Electromagnetics 3D Parallel) is a suite of codes for solving for electromagnetic fields in complex geometries [1]. It is based on high-order curved finite elements, and implemented in C++ using MPI routines for parallel processing. ACE3P is structured as a toolkit with different modules, listed in Fig. 1. ACE3P workflow consists of generating a geometry in Cubit [2], converting the Cubit output file into NetCDF form using the processing tool acdtool, running ACE3P, and post processing in a software like Paraview [3], as shown in Fig. 2.

ACE3P (Advanced Computational Electromagnetics 3P)

Frequency Domain:	Omega3P	– Eigensolver (damping)
	S3P	– S-Parameter
Time Domain:	T3P	– Wakefields and Transients
Particle Tracking:	Track3P	– Multipacting and Dark Current
EM Particle-in-cell:	Pic3P	– RF guns & space charge effects
Multi-physics:	TEM3P	– EM, Thermal & Mechanical analysis
Static Particle-in-cell:	Gun3P	– DC guns & space charge effects

Figure 1: ACE3P modules.

For this paper, we will focus on the S3P module. S3P calculates the scattering parameters of RF components by solving Maxwell's equations in the frequency domain. S3P supports dielectric and ferric materials and offers absorbing and PML boundary conditions.

LUME-ACE3P

LUME-ACE3P (LUME: Lightsource Unified Modeling Environment) is a set of Python code interfaces for running ACE3P workflows (including Cubit and postprocessing routines) with the intent of running parameter sweeps or optimization problems [4]. The base structure of LUME-ACE3P is built on LUME [5]. LUME-ACE3P supports the S3P and Omega3P modules in ACE3P. To perform a parameter sweep or optimization run, a user needs to provide the following:

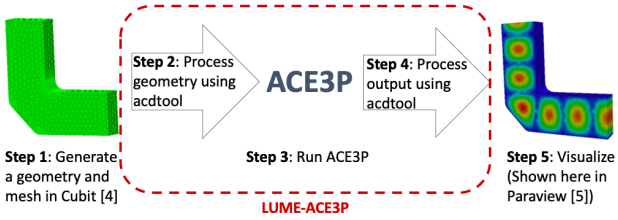

Figure 2: ACE3P workflow with LUME-ACE3P workflow overlaid.

- a LUME-ACE3P configuration file (.yaml) containing the workflow settings, input/output parameters, and ACE3P settings (this may also go in the optional ACE3P input file)
- a Cubit journal (.jou) file for editing (required for remeshing)
- a batch script (.batch) for submitting a job to the appropriate HPC resources
- (OPTIONAL) an acdtool postprocess file (e.g. .rfpost) with desired postprocessing settings (only needed for Omega3P)
- (OPTIONAL) an ACE3P file (.s3p or .omega3p) for setting ACE3P parameters like boundary conditions (if not included, this information must be specified in the LUME-ACE3P configuration file)

Before this result, running optimization with LUME-ACE3P would have required trial and error. While running a parameter sweep with LUME-ACE3P requires little time in set up from the user, it is a computationally expensive way to solve a problem like optimization that can be done in much fewer steps.

Xopt

Xopt is a Python package for optimization [6]. It supports twelve different optimization algorithms of genetic, Bayesian and deterministic types. To run an optimization problem, a user needs to specify a VOCS (variables, objectives, constraints) dictionary object and a sim function. The sim function takes input parameters and returns the quantity

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-WEP029

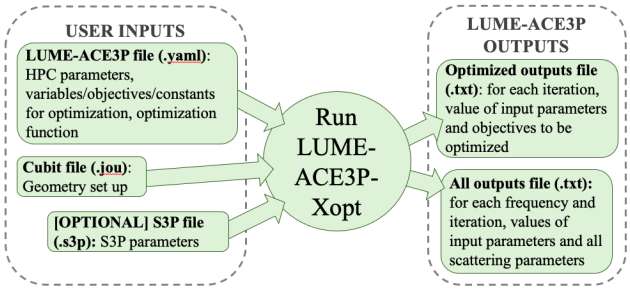

Figure 3: Input and output file structure for using LUME-ACE3P with Xopt.

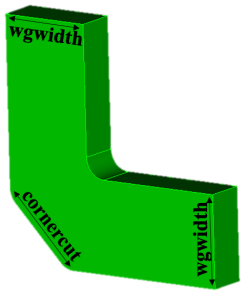

Figure 4: Waveguide with 90 degree bend, with input parameters of chamfer length ('cornercut') and waveguide width ('wgwidth') labeled.

to be optimized. Running Xopt directly requires familiarity with Python.

LUME-ACE3P AND XOPT

We have integrated Xopt with LUME-ACE3P workflow in the S3P module for user-friendly optimization. To perform an optimization, the user must include a "vocs_dict" in the LUME-ACE3P configuration file specifying variables, objectives, and constraints, as well as an "xopt_dict" specifying what kind of optimization evaluator is to be used and tolerances/number of steps. There are two output files, one with user specified input and one with only data for optimized parameters. The input and output file structure is shown in Fig. 3.

LUME-ACE3P with Xopt is configured for the S3P module. It supports multi-objective optimization and allows users to choose between Nelder-Mead and Expected Improvement optimization algorithms, with more choices for algorithm in development.

LUME-ACE3P EXAMPLE

Set Up

To showcase the code's optimization capabilities, we present a toy example using LUME-ACE3P to minimize reflections in a rectangular waveguide with a 90 degree bend. We let the chamfer length ('cornercut') and waveguide width ('wgwidth') be input parameters, as in Fig. 4.

We use S3P to find the scattering parameters in this cavity, specifically seeking to minimize S(1,1) at the frequency 12 GHz. We use the Nelder-Mead optimization algorithm

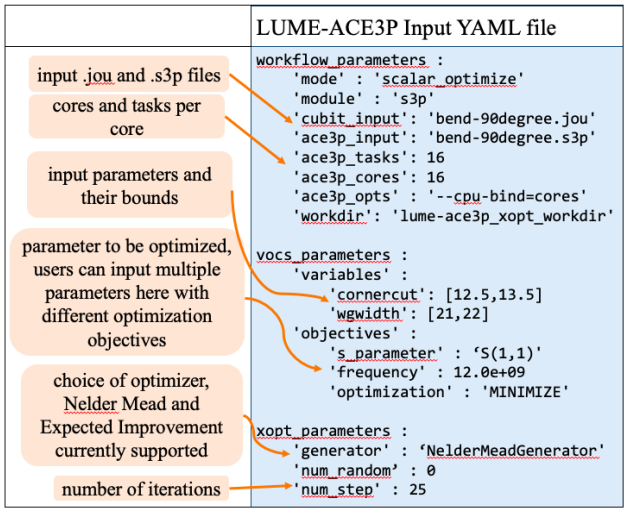

Figure 5: Example input YAML file for running LUME-ACE3P with Xopt.

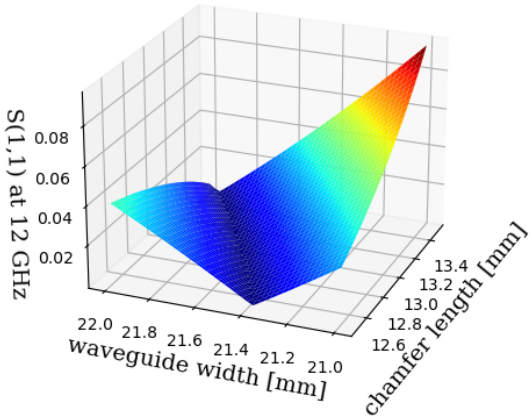

Figure 6: S(1,1) at 12 GHz as a function of the two input parameters. The color scale matches that of Fig. 7.

with 25 steps. All of the information about the optimization parameters can be set up in a single LUME-ACE3P configuration file, shown in Fig. 5.

Results

LUME-ACE3P with Xopt outputs two files, one containing results only for the optimized S-parameters and frequencies, and one containing all scattering parameters and frequencies. We ran a sweep of the optimization parameter space in addition to this optimization run in order to understand the behavior of the optimizer.

Figure 6 shows S(1,1) at 12 GHz plotted in three dimensions as a function of the two input parameters. The colors used in this plot indicate the magnitude of S(1,1), with blue being closer to 0. We expect the optimization algorithm to find the dark blue valley.

Figure 7 shows the optimization algorithm's choice of points overlaid on a color map of S(1,1) at 12 GHz (the colors match those in Fig. 6). From these results, we found that the Nelder-Mead optimizer reached within 7% of the

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-WEP029

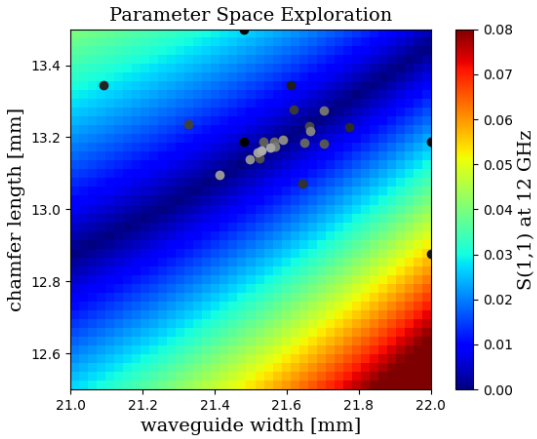

Figure 7: The optimizer's choice of points overlaid on a color map of S(1,1) at 12 GHz. The points vary in color from black to light gray as iteration number increases.

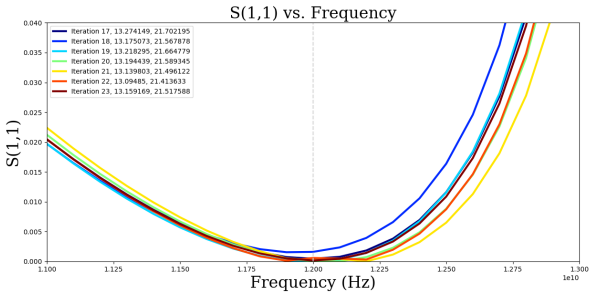

Figure 8: S(1,1) plotted as a function of frequency for the last seven iterations.

true minimum in 25 steps. The advantage of having an optimization method is evident—finding the minimum through a parameter sweep would require a large number of steps scaling with parameter space dimension and parameter sweep resolution.

Plotting

LUME-ACE3P offers two built in optimization plotting tools. Figure 8 shows S(1,1) as a function of frequency, plotted for the last seven iterations. Users can choose which iterations appear in this plot.

Figure 9 is a still shot from a dynamic plotting tool that allows users to plot S-parameters of interest as a function of frequency with a slider for S-parameter and iteration number.

CONCLUSION AND FUTURE WORK

LUME-ACE3P has been integrated with optimization for the S3P module. Optimization with LUME-ACE3P is easy to use, as a user only needs a Cubit file and a LUME-ACE3P configuration file to run a problem. It is also geared to real applications: users can match waveguide impedances for accelerator component design by optimizing over one or more scattering parameters, and users can choose between Nelder-Mead and Expected Improvement optimization. Finally, optimization with LUME-ACE3P is set up for simple visualization, as built-in tools allow for clear exploration of results.

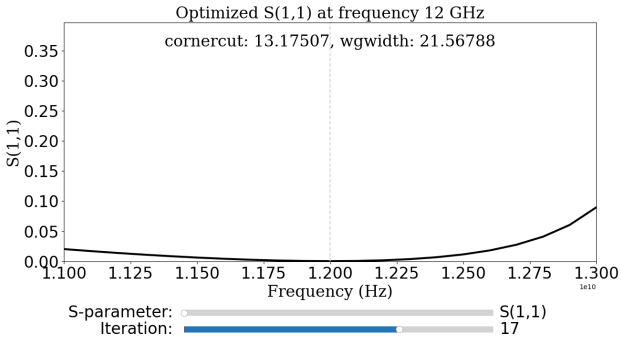

Figure 9: S parameter at iteration number of choice plotted as a function of frequency.

Future work on LUME-ACE3P with Xopt includes expanding the number of optimization algorithms offered and adding optimization for Omega3P. Check in with our Github for the latest updates [4].

ACKNOWLEDGEMENTS

This work was supported by US Department of Energy under contract AC02-76SF00515 and the Department of Energy Science Undergraduate Laboratory Internships program. This research used resources of the National Energy Research Scientific Computing (NERSC) Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] C.-K. Ng *et al.*, "Advances in Parallel Finite Element Code Suite ACE3P", in *Proc. 6th Int. Particle Accelerator Conf. (IPAC'15)*, Richmond, VA, USA, May 2015, pp. 702-704. doi:10.18429/JACoW-IPAC2015-MOPMN002

[2] T. D. Blacker *et al.*, *CUBIT Geometry and Mesh Generation Toolkit 15.2*, SNL, Albuquerque, NM, USA, May 2016. doi:10.2172/1457612

[3] J. Ahrens, B. Geveci, and C. Law, *ParaView: An End-User Tool for Large Data Visualization, Visualization Handbook*. New York, NY, USA: Elsevier, 2005.

[4] D. Bizozzero and L. Fowler, "LUME-ACE3P," GitHub, August 2025, https://github.com/slaclab/lume-ace3p.git

[5] C. E. Mayes *et al.*, "Lightsource Unified Modeling Environment (LUME), a Start-to-End Simulation Ecosystem", in *Proc. 12th Int. Particle Accelerator Conf. (IPAC'21)*, Campinas, SP, Brazil, May 2021, pp. 4212-4215. doi:10.18429/JACoW-IPAC2021-THPAB217

[6] R. Roussel *et al.*, "Xopt: A simplified framework for optimization of accelerator problems using advanced algorithms", in *Proc. 14th Int. Particle Accelerator Conf. (IPAC'23)*, Venice, Italy, May 2023, pp. 4847-4850. doi:10.18429/JACoW-IPAC2023-THPL164