Automation of ANAELU 2D-XRD texture analysis refinement through Intelligent Computing

Juan Fernando de la Cruz de la Cruz[1] , Edgar Eduardo Villalobos Portillo[2], Luis Fuentes Montero[3], Luis E. Fuentes Cobas[2]

[1]University of the Ozarks 415 College Ave. Clarksville, Arkansas, USA, [2] Centro de Investigación en Materiales Avanzados, Miguel de Cervantes 120. Complejo Industrial Chihuahua, Chih. México, [3]Diamond Light Source Ltd., Harwell Science and Innovation Campus, Didcot OX11 0DE, UK

jdela757@ozarks.edu, edgar.viilabolos@cimav.edu.mx, luis.fuentes-montero@diamond.ac.uk, luis.fuentes@cimav.edu.mx

Introduction

Recently developed x-ray 2D detectors generate the demand for fast and reliable tools for 2D-XRD patterns interpretation. The CIMAV Crystallography Group is active in this field, having introduced the novel software package ANAELU. This Rietveld-style program allows the characterization of polycrystal structure, particularly texture evaluation, by means of 2D-XRD modeling and fitting to experimental data. Previous versions of ANAELU worked accurately, but, due to the manual character of the parameters optimization, its application represents a slow process. By means of the present work the ANAELU group enters a renovation process for the program, introducing the automatic-fitting option for various parameters of the model pattern.

ANAELU

The Analytical Emulator Laue Utility (ANAELU) package models diffraction patterns from single crystals and from textured polycrystals through the use of mixed programming among compiled (FORTRAN 95/2003) and Interpreted languages (Python). The user interacts with the Python GUI in WxPython that is programmed in such a way it can read diffraction patterns through the FabIO library. The calculation of diffraction patterns are performed through the systematic use of the CRYSFML library through FORTRAN 2003.

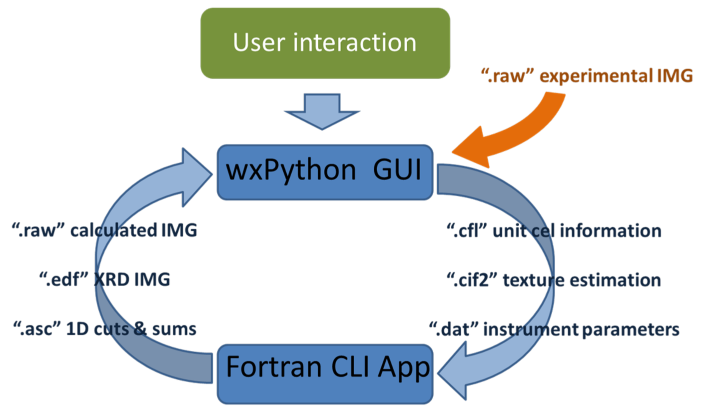

Figure 1. ANAELU Working Interaction.

ANAELU Modeling Cycle

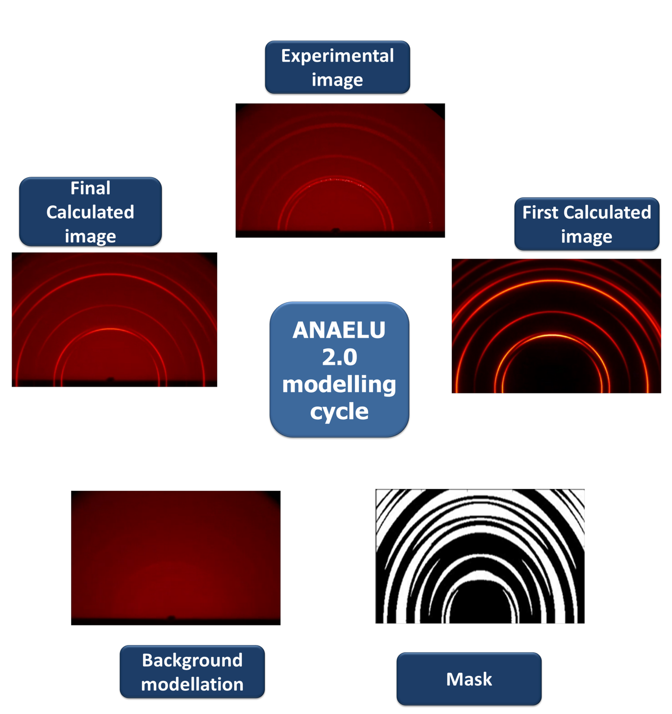

Figure 2. Global view of computer-aided interpretation of a 2D-XRD pattern.

Visual Representation of ANAELU refinement

Visual Results of Algorithm Performance

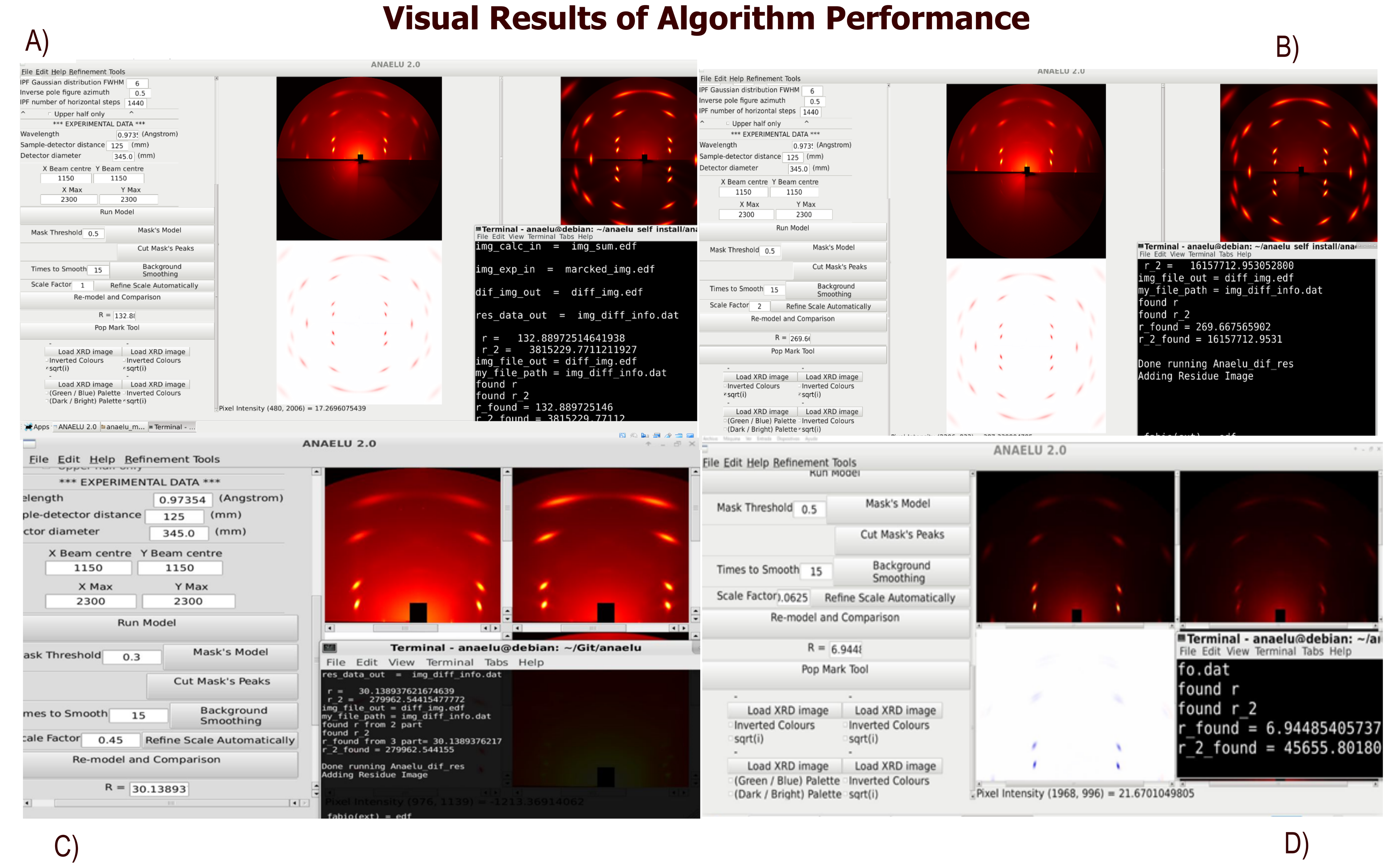

Figure 3. **A)** Zeroth Cycle, **B)** First Cycle, **C)** Third Cycle, **D)** Fifth Cycle.

Optimization Method

The technique selected for the parameters optimization is the Nelder-Mead(NM) simplex algorithmic method. This algorithm was adapted to the Python GUI of ANAELU. Given that ANAELU can receive different values for parameters such as the scale factor, IPF Gaussian value, average crystal size, etc., and give in return an R value that measures the difference between the experimental 2D-XRD image and the calculated image it was clear that any optimization algorithm used for adding a refinement setting to ANAELU must be an algorithm that can optimize a “black box.” That is, an algorithm able to optimize an objective function f without having any information about the gradient or whether the function is continuous or differentiable. Thus, the technique selected for this process was the Nelder-Mead algorithm method, an optimization method known as a fast an widely used algorithm in local minimum optimization problems. The following diagram gives information about the functioning of such algorithm .

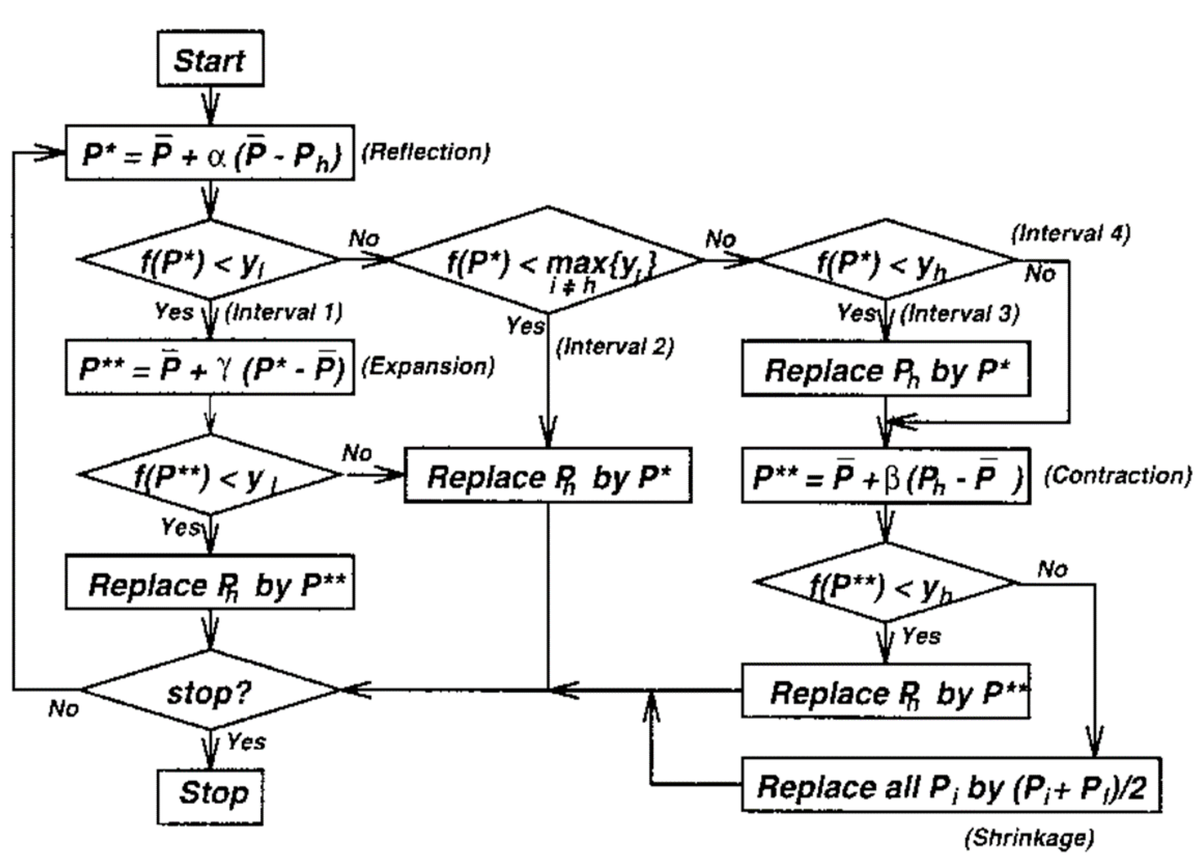

Figure 4. Flowchart for downhill simplex search.

Results and Conclusions

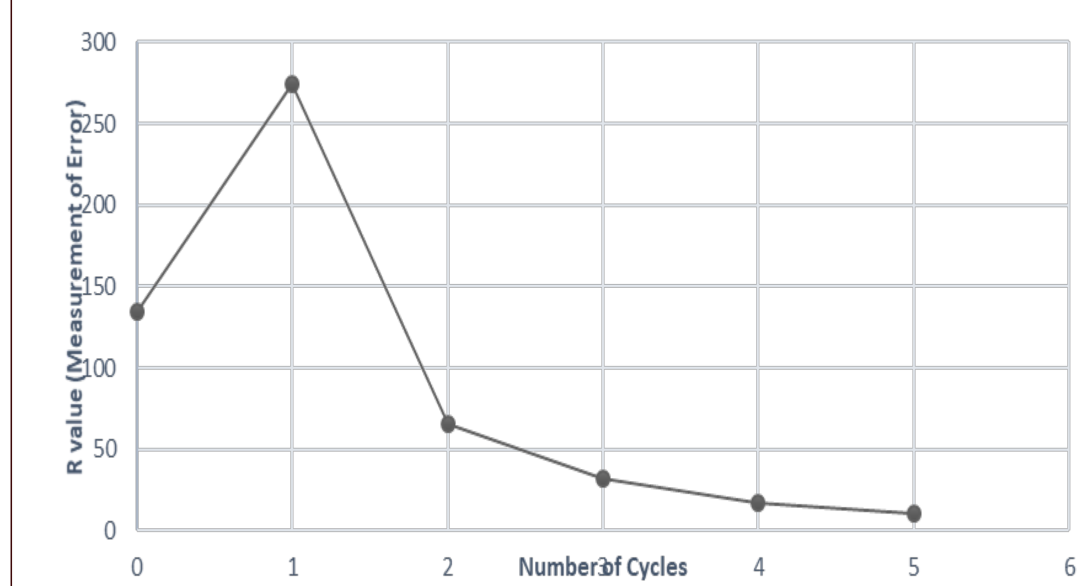

Figure 5. Algorithm Performance in ANAELU.

- The implementation of the Nelder-Mead algorithm in ANAELU showed to have a positive impact during the refinement process.
- The automatic fitting process runs fast and stably
- By means of the current contribution, ANAELU gets significantly close to becoming the first 2D Rietveld – oriented texture analysis program

Acknowledgements

Thank you to my advisor Dr. Luis Fuentes, and all these who have supported me furthering my research experience through feedback and presentation opportunities, specially to Eduardo who always supported me and gave so much of his time to sit down with me an explain several concepts and helping in the hardest part of the project

Works cited

- Agrawal, S., & Singh, D. (2017). Modified Nelder-Mead self organizing migrating algorithm for function optimization and its application. Applied Soft Computing, 51, 341-350.
- Luersen, M. A., & Le Riche, R. (2004). Globalized Nelder–Mead method for engineering optimization. Computers & structures, 82(23-26), 2251-2260.
- Introduction to Intelligent Computing: Downhill Simplex Search. (n.d.). Retrieved from http://www.cs.nthu.edu.tw/~jang/courses/cs4601/simplex.html (Fig 4).
- Burciaga-Valencia, D.C., Villalobos-Portillo, E.E., Marín-Romero, J.A. et al. J Mater Sci: Mater Electron (2018) 29: 15376. https://doi.org/10.1007/s10854-018-8919-1 (Fig 1 and 2).