First Release: January 7, 2024 This Release: April 28, 2024

IntegralOpt User Guide

Version 2a

Steve Kahn Ramesh Gupta

History: The program, now called IntegralOpt, was originally written for VAX fortran by Ramesh Gupta (BNL) in 2004 for optimizing the optimum integral design. The optimum integral design was proposed by Ramesh Gupta in 2004 and used in designing and building a corrector dipole for AGS Helical magnet. The program was originally called MINXEND (for optimizing <u>X</u>-section and <u>END</u> design together, as per the optimum integral design approach, <u>MIN</u> refers to the use of CERN MINUIT routines for optimization). The program has now (2021-2024) been transported to Linux PC (first on Cygwin and now on Ubuntu) by Steve Kahn (now at PBL, was earlier at BNL) and by Ramesh Gupta as a part of Phase I and Phase II STTR and is being periodically updated. The manual was originally written by Steve Kahn.

Introduction

The IntegralOpt program is designed to provide the maximum integral field while minimizing the unwanted field harmonics. The program finds the optimum number of turns per coil block, wedge angle between blocks, and turn spacing within a block in both the magnet straight and end sections. The program uses the MINUIT [1] minimization program to the determine the best parameters consistent with minimizing higher order harmonics. Once determined the parameters are used to produce a coil representation that can be used as an input file for the OPERA [2] program. The program produces an OPERA input files that represent coils by their individual turns or by individual coil blocks. The individual turn representation is more precise than the block representation but requires more computer time in the analysis. The OPERA program using the IntegralOpt coil files are used to obtain the peak field on the coils.

The IntegralOpt program was written in Fortran 90/95 and was developed using the gfortran compiler.

Installation

The program is distributed as a zip file that contains the fortran code and a bash script to compile and link the code. It uses the minuit package from CERN. A minuit.a library constructed on ubuntu is provided. If the user uses a different platform, he will need build his own minuit library.

The program works well with Ubuntu, an open-source LINUX type operating system. We have installed it on PC running WINDOWS operating system. Zip files should be extracted and copied to a new directory via WINSCP or equivalent. Remember to perform CHMOD or equivalent to make .sh files executable (x). BUG ALERT: Please do ./build-minx1.sh twice to create executable minx.exe

The bash shell build file is shown:

#/bin/bash gfortran -c -g -O0 current.f gfortran -c -g -O0 garc.f qfortran -c -q -O0 sbrxyz.f gfortran -c -g -O0 gbrick.f gfortran -c -g -O0 cde_declarations.f gfortran -c -g -O0 nmlis_declar.f gfortran -c -g -O0 cons.f gfortran -c -g -O0 blocks.f gfortran -c -g -O0 brick.f gfortran -c -g -O0 readopt.f gfortran -c -g -O0 cnbnd.f gfortran -c -g -O0 xend_init.f gfortran -c -g -O0 gstraight.f gfortran -c -g -O0 wire_mode.f gfortran -c -g -O0 brick_mode.f

gfortran -c -g -O0 MINXENDK5.f

gfortran -g -O0 MINXENDK5.o cde_declarations.o blocks.o brick_mode.o current.o garc.o gbrick.o gstraight.o nmlis_declar.o wire_mode.o xend_init.o cnbnd.o readopt.o brick.o cons.o sbrxyz.o -o minx.exe -L. -lminuit

To build the executable one needs to type (bug alert: do the following operation twice) ./build_minx1.sh

The required inputs to run the program are file.X01 and file.X07 files where "file" is the name given to the project. To run the executable: ./minx.exe file

Input Files

There are two input files that must be supplied to run the program. These are "file.X01 and file.X07" where "file" is the file name specified on the execution line.

ALERT: In some installations, "X" in filenames "file.X01" and "file.X07" must be in uppercase.

File.X01

An example of the X01 input file is shown:

\$FCNX VC2CB=.TRUE.,VC2CE=.TRUE.,MAGTYPE=2,LAYERS=6,RFEMM=110,R0MM=38., RBENDMM=10,NBEND=10 &end 3 3 0.6 1.1 58.06 500 0.4 0.20 3 3 0.6 1.1 59.28 500 0.4 0.20 2 2 0.6 1.1 64.03 500 0.4 0.20 2 3 0.597 1.1 65.25 500 0.4 0.20 3 3 0.6 1.1 68.28 500 0.4 0.20 3 3 0.6 1.1 69.49 500 0.4 0.20 B2 0. 1. B4 0. 2. b6 0. 5. b8 0. 20. b10 0. 20. b12 0. 9.

in table 1. The next several lines give parameters describing each layer. There is one such line for each layer. The layer line which consists of the input variables is shown in table 2: All inputs on the layer line must be supplied.

The last section of the X01 file specifies the harmonics and the associated weights used in the optimization.

Table 1: NAMELIST FCNX input variables
--

Variable	Units	Default Value	Description		
MAGTYPE		2	2 for Dipole; 4 for quadrupole; 3 for combine functior		
ROMM	mm	1.0	Reference Radius in mm		
LAYERS		1	Total Number of Layers		
RFEMM	mm	0	Iron Inner Radius in mm; If zero, No Iron		
RBENDMM	mm	0	Bend Radius at Transition to End Sector		
NBEND		10	Number of Points on Bend Curve		
VC2CB*		.FALSE.	Flag to allow variable conductor spacing in body		
VC2CE*		.FALSE.	Flag to allow variable conductor spacing in ends		
ENDCONFIG [*]		2	0 for Straight Section; 1 for ½ body plus 1 end; 2		
			optimum integral end; 3 for serpentine [*] w/ SS		
TAPER*		.FALSE.	Flag to indicate that the coils are tapered		
TAPERTYPE [*]		0	?		
MAXANGLE	degrees	0	Angle subtended at which a block is divided		

*Features not yet fully implemented or tested

Table 2: Variables on Layer Line

Layer Variable	Units
Number of Blocks in Layer Straight Section	
Number of Blocks in Layer End Section	
Straight Section Length	meters
Wire Diameter	mm
Layer Coil Radius	mm
Wire Current	A
Wire Insulation in End Section	mm
Wire Insulation in Straight Section	mm

File.X07

The X07 file is input to MINUIT. It specifies the fit parameters, initial values, and range of parameters during the fit. Shown below is a typical input X07 file. Specifically, for each parameter there is the parameter number, parameter name, initial value, initial weight, and lower and upper limits. There are three parameters for each block in the straight section and in the end section in each layer. The parameters for a block in the straight section are the start angle (Wn), the number of wires in the block (Nn), turn-to-turn conductor spacing (Bn). In the end region the parameters are the end blocks (Sn), number of end turns (Tn), and end turn-to-turn conductor spacing (En). There are six parameters for each block in each layer. A large number of parameters are required for a case with many layers. Minuit as supplied by CERN has a limit of 100 parameters. This may not be enough for a design with ten layer. The CERN code can be modified and recompiled to handle a larger number of parameters. The current version has been updated for 200 parameters and has been used in designing a 12-layer coil.

The last several lines are input to MINUIT to control the fit procedure.

MINxend: 3,3 BODY & 3,3 ENDS, 2 LAYERS, (CONSTANT VC2CB & VC2CE)

1 W1	0	0	0	19
2 N1	38	0.	20	40
2 R1	0	0	0	9 9
4 W/2	2	3 48	1	э. Д
5 N2	20	0	10	20
6 B2	0	0.	0	9 9
7 W3	о. Л	0. 1 81	0.	J. 5
8 N3	-т. Л	ч.04 Л	י. כ	5
9 B3	ч. 0	0.	0	9. 9
10 51	0.	0.0	0.	20
10 51 11 T1	30	0.0	о. Л	20.
12 F1	0	0.0	ч. 0	10
12 52	2	0.0	0.	11
13 32 17 T2	2.	0.	0.	29
14 12 15 E2	20. 0	0.0	0.	2J. 10
16 53	2.5	0.0	0.	10.
10 33 17 T3	2.J 1	0.	0.	79 29
17 13 18 E3	4. 0	0.	0.	2J. 10
10 LJ	0.	0.0	0.	10.
20 N/	0. 29	0.	0. 20	19.
20 N4 21 B4	2J. 0	0.	0	40. Q
27.04	0.	2.88	0.	J. 5
22 WJ 23 N5	1. 11	2.00 0	5 0.	20
23 NJ 24 R5	0	0.	0	20. a
24 DJ 25 W6	0.	0.82	0.	ן. כ
25 W0	۱. 5	0.02	л Л	15
20 NO 27 B6). 0	0.	0	٦ <i>٦</i> .
28 54	0.	0.0	0.	J. 10
20 34 29 T4	0. 25	0.0	о. Д	29
20 F4	0	0.0	ч. 0	10
31 55	2.5	0.0	0.	10.
37 35 32 T5	10	0.	0.	29
32 F5	0	0.0	0.	10
34 56	25	0.0	0.	10.
35 T6	10	0.	0	29
36 F6	0	0.0	0.	10
50 20	0.0	0.0	0.	10.
CALL FON	2.0 2			
SIMPLEX	5.			
	3			
MIGRAD	5.			
	3			
MINIMI7F	5.			
	3 1			
END RETURN	J. 1.			
	•			

Output Files

There are many useful output files that are provided for different uses:

File.x91:

This file provides single wire conductor description that can be input into opera.

File.x81:

This file provides a brick-20 conductor description that can be input into opera.

File.82:

This file provides a brick-8 conductor description that can be input into opera. This file is NOT recommended.

File.11:

Summary of design parameters obtained in the analysis.

File.31:

This file provides a turn-by-turn description of the coil. This description provides the radius, angle, turn length, x-position, y-position of each turn for each layer.

APPENDIX: README FILE (README.txt in the extracted package)

Instructions for building and executing OptIntegral_v2:

Unpack into some directory

./build_minx1.sh This builds the minx.exe

Required inputs:

L1cs.X01 and L1cs.X07

Note that there is a new namelist parameter MAXANGLE. If not present it should run as before.

If present it specifies the angle in degrees where the block is split.

./minx.exe L1cs This runs the executable. L1cs is the name for input and output files.

This version includes:

- -- The ability to split a block such that the maximum angular width is less than a specified amount. There is a namelist parameter MAXANGLE to set the angular width.
- -- All coils adjacent to the pole are GARD instead of BRICK20. (This appears to work, but it produces a lot of conductor elements and still is slow)
- -- A pre-check is made on all transition BRICK20 elements to see if they violate the 25%/75% mid-length test preformed by Opera version 20 and above. If it fails no conductor is passed to OPERA. Only one case was failed at layer 6 in both the 6 layer and 10 layer models.

As mentioned OPERA is still too slow. It takes ~24 hours to process the 6 layer model.