MAGNET SHIMMING

An IBM-7090 Code Utilizing Empirical Data to Obtain Optimal Shimming and Trimming of Magnets

by

Nils Vogt-Nilsen

G E N E V A
Propriété littéraire et scientifique réservée pour tous les pays du monde. Ce document ne peut être reproduit ou traduit en tout ou en partie sans l’autorisation écrite du Directeur général du CERN, titulaire du droit d’auteur. Dans les cas appropriés, et s’il s’agit d’utiliser le document à des fins non commerciales, cette autorisation sera volontiers accordée.

Le CERN ne revendique pas la propriété des inventions brevetables et dessins ou modèles susceptibles de dépôt qui pourraient être décrits dans le présent document ; ceux-ci peuvent être librement utilisés par les instituts de recherche, les industriels et autres intéressés. Cependant, le CERN se réserve le droit de s’opposer à toute revendication qu’un usager pourrait faire de la propriété scientifique ou industrielle de toute invention et tout dessin ou modèle décrits dans le présent document.

Literary and scientific copyrights reserved in all countries of the world. This report, or any part of it, may not be reprinted or translated without written permission of the copyright holder, the Director-General of CERN. However, permission will be freely granted for appropriate non-commercial use.

If any patentable invention or registrable design is described in the report, CERN makes no claim to property rights in it but offers it for the free use of research institutions, manufacturers and others. CERN, however, may oppose any attempt by a user to claim any proprietary or patent rights in such inventions or designs as may be described in the present document.
MAGNET SHIMMING

An IBM-7090 Code Utilizing Empirical Data to Obtain Optimal Shimming and Trimming of Magnets

by

Nils Vogt-Nilsen

G E N E V A
TABLE OF CONTENTS

Page

Introduction .............................................. 1
Theory and Method ....................................... 2
Programme Input Data .................................. 7
Programme Output ....................................... 13
Other Applications ..................................... 14
MAGNET SHIMMING

An IBM-7090 Code Utilizing Empirical Data to Obtain Optimal Shimming and Trimming of Magnets

by

Nils Vogt-Nilsen

INTRODUCTION

The frequently encountered problem of producing a specified magnetic field within a given volume is rarely completely solved by the initial design of a magnet. Depending on the accuracy wanted in the result, it is in the majority of cases necessary to correct the more or less rough field of the magnet by introducing appropriate shims or trimcoils. For narrow tolerances in the required field the necessary number of such correcting elements frequently becomes so large that their adjustment by pure experience and cut-and-try methods will be exceedingly tedious and the solution attained seldom the best possible.

At any stage in the course of a shimming or trimming process the subsequent corrections to be tried are based on the mass of empirical data previously accumulated. The analysis of such data becomes increasingly complex for larger number of correcting elements. It is the object of the computer programme to be described in the following to perform this analysis in such a manner that one may always obtain the optimal field corrections for a chosen number of correcting elements.
THEORY AND METHOD

The effects on the magnetic field resulting from variations in the correcting elements are observed at a chosen set of checkpoints distributed throughout the region of interest. There need be no relation between these checkpoints. They may be chosen completely at will, for instance along a curve, on a surface, or within a volume. The only requirement is that the same checkpoints are used throughout the correction process.

The magnetic field at the $m^{th}$ checkpoint is then a function,

$$B^m = B^m(p_1, p_2, \ldots, p_n), \quad (1)$$

of the set of parameters $p_1, p_2, \ldots, p_n$, which are to be varied during the correction process. Such parameters may be typically the main coil current or currents, trim coil currents, geometrical quantities defining the shapes and positions of trim coils and magnet shims, etc. It is assumed that for each checkpoint the function (1) may be expressed with sufficient accuracy as a polynomial

$$B^m = c_0^m + \sum_{i=1}^{n} c_i^m p_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^m p_i p_j + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} c_{ijk}^m p_i p_j p_k + \cdots \quad (2)$$

truncated after a specified number of terms $N$.

For computational simplicity these polynomials are expressed

$$B^m = \sum_{i=1}^{N} D_i^m Q_i, \quad (3)$$

the terms being identical and taken in the same order as above; thus for each checkpoint $m$: 
\[ D_1^m = C_0^m \quad Q_1 = 1 \]
\[ D_2^m = C_1^m \quad Q_2 = P_1 \]
\[ D_3^m = C_2^m \quad Q_3 = P_2 \]
\[ \vdots \]
\[ D_{n+1}^m = C_n^m \quad Q_{n+1} = P_n \]
\[ D_{n+2}^m = C_{11}^m \quad Q_{n+2} = P_1 \]
\[ D_{n+3}^m = C_{12}^m \quad Q_{n+3} = P_1P_2 \]
\[ D_{n+4}^m = C_{13}^m \quad Q_{n+4} = P_1P_3 \]
\[ \vdots \]

The programme contains a special subroutine for producing the \( Q_1, Q_2, \ldots, Q_N \).

The coefficients of the polynomials (3) may be determined from empirical data provided that the field at each checkpoint has been measured for a sufficient number (at least \( N \)) of independent settings of the parameters \( P_1, P_2, \ldots, P_N \). It has been chosen to do this by a least squares fit,

\[ \delta \sum_{\text{variations}} (B_{\text{measured}}^m - B^m)^2 = 0, \quad (5) \]

the sum to be taken over all parameter variations. By introducing the polynomials (3) and varying their coefficients \( D_{ij}^m \), one obtains for each checkpoint \( m \) a set of \( N \) linear algebraic equations

\[ \sum_{i=1}^{N} \left( \sum_{\text{variations}} Q_i Q_j \right) D_{ij}^m = \sum_{\text{variations}} B_{\text{measured}}^m Q_j, \quad (j = 1, 2, \ldots, N) \quad (6) \]

for the determination of these coefficients.
It is noted that in the Eqs. (6) the system matrix, and hence also the triangularizing process, is common for all checkpoints, whereas the right hand side vector differs from one checkpoint to the next. Thus, the same Eqs. (6) must be solved once for each checkpoint.

Observe also that if the number of parameter variations exceeds the chosen number of terms \( N \), the least squares polynomial fit ensures a certain amount of smoothing of the empirically obtained data.

The object of the whole correction process by shimming, trimming, etc., is to bring the field as close as possible to given desired values at each of the checkpoints. It is here chosen again to use a least squares method, whereby the parameters \( p_i \) are determined such that the sum over all checkpoints of the squared differences between the obtained fields and the desired fields is at a minimum. Hence, for \( M \) checkpoints

\[
\delta \sum_{m=1}^{M} \left( E^m_{\text{observed}} - E^m_{\text{desired}} \right)^2 = 0, \tag{7}
\]

where the parameters \( p_1, p_2, \ldots, p_n \) are to be varied, and the fields \( E^m \) should be inserted by the polynomials (2) or (3), whose coefficients now have been determined by the Eqs. (6). Accordingly the optimal parameters are determined by the \( n \) equations

\[
\sum_{m=1}^{M} \left( \sum_{i=1}^{N} E^m_{i} Q_i - E^m_{\text{desired}} \right) \sum_{j} E^{mk}_{j} Q_j = 0, \quad (k = 1, 2, \ldots, n) \tag{8}
\]

where

\[
\frac{\partial E^m}{\partial p_k} = \sum_{j} E^{mk}_{j} Q_j, \quad (k = 1, 2, \ldots, n) \tag{9}
\]

are again polynomials of exactly the same type as (2) or (3), but of one degree lower. To each term in (2) which involves the parameter \( p_k \) there is a corresponding term in (9), Thus, the number
of terms depends on \( N \) and \( k \). Similarly, from (9) one may develop the second order partial derivation polynomials

\[
\frac{\partial^2 E^m}{\partial p_i \partial p_k} = \sum_{j} F_{j}^{mk} Q_{j} , \quad (i,k = 1,2,\ldots,n), \quad (10)
\]

Excepting the case when the polynomials (2) are linear in the parameters \( p_1, p_2, \ldots, p_n \), the Eqs. (8) will be non-linear. In the programme their solution is found by the method of steepest ascent: Given any approximate solution \( p_1^*, p_2^*, \ldots, p_n^* \), a better solution may be determined by expanding the Eqs. (8) in Taylor series, retaining only linear terms, and finding the corrections \( \Delta p_1, \Delta p_2, \ldots, \Delta p_n \) by the following set of \( n \) linear equations:

\[
\sum_{m=1}^{M} \left( \sum_{i=1}^{N} D_{i}^{m} Q_{i} - B_{\text{desired}}^{m} \right) E_{j}^{mk} Q_{j} + \sum_{\alpha=1}^{n} \left( \sum_{m=1}^{M} \left[ \left( \sum_{i=1}^{N} D_{i}^{m} Q_{i} - B_{\text{desired}}^{m} \right) E_{j}^{mk} Q_{j} \right] \right) \Delta p_{\alpha} = 0, \quad (k=1,2,\ldots,n). \quad (11)
\]

The corrected solution to the Eqs. (8) is then \( p_i + \Delta p_i \), \( (i = 1,2,\ldots,n) \). The convergence by iterated use of the Eqs. (11) is very rapid.

Having by these means determined the optimal parameter settings \( p_1^*, p_2^*, \ldots, p_n^* \), the final field values and residues are given by insertion into the polynomials (3):

\[
E_{\text{final}}^{m} = \sum_{i=1}^{N} E_{i}^{m} Q_{i} , \quad (m = 1,2,\ldots,M) \quad (12)
\]

\[
E_{\text{residue}}^{m} = E_{\text{final}}^{m} - E_{\text{desired}}^{m}.
\]
The accuracy of these results must now be checked empirically. It may well happen that the first set of measured data is not comprehensive enough, or that it only badly covers the optimal solution. In the latter case one might have been extrapolating rather than interpolating on one or several of the parameters. In such cases a second set of smaller parameter variations around the obtained solution should quickly lead to consistent results.

In this connection one might follow one of several procedures, depending entirely on the case at hand. If one is sufficiently close to the correct solution it is enough to use a linear function (2) for the second set of measurements. In fact one might choose to work entirely with a sequence of linear approximations (2), each based on an independent set of measurements, and in this manner successively approach the optimal solution.

On the other hand one might extend the first set of measurements with the second set, thereby giving more information close to the optimal solution. One is then also able to include more terms in the polynomial approximation (2), (3), and thereby obtain more exact results.

Finally one may decide to give the second set of measurements double or higher weight in the least squares fit (5), (6) by presenting this data twice or more to the computer. This would be appropriate if the second set of measurements is performed with a higher accuracy than the initial set.

The optimal corrections are found when the computed final fields (12) are produced correctly in the magnet. If the accuracy given by the field residues at the checkpoints is less than required, this indicates that the number of parameters or correcting elements has not been chosen large enough, and the procedure must be repeated with additional parameters introduced.
PROGRAMME INPUT DATA

The IBM-7090 programme to be described works under Fortran II Monitor control and reads data cards of 80 columns through the IBM-1401. All numerical data is furnished in 10-column number fields; thus a maximum of 8 numbers may be read in per card. On the first page of output the input data pertaining to a complete case is listed for checking purposes. The 10-column subdivision facilitates card punching and makes the input listing easily surveyable.

Nine groups of data are necessary to initiate the calculation. The leading card (if more than one) in each group contains a codeword starting in column 1 and not exceeding the first 10 columns. The programme will compute any number of cases in succession, and will remember data groups from one case to the next in order to avoid unnecessary repetitions of input data.

On page 8, the various data groups, with their codewords underlined, are listed schematically; followed subsequently by a more precise description.
IDENT  Text identifying the case

PARAMETERS  n  (≤ 20)

CHECKPTS  M  (≤ 100)

DEGREE  J

POLYNOM  N  (≤ 100)

VARIATIONS

\[
\begin{align*}
P_1 & \quad P_2 & \quad P_3 & \quad \ldots & \quad P_n \\
P_1^{\text{measured}} & \quad P_2^{\text{measured}} & \quad P_3^{\text{measured}} & \quad \ldots & \quad P_M^{\text{measured}}
\end{align*}
\]

(may be repeated)

DESIRE

\[
\begin{align*}
P_1^{\text{desired}} & \quad P_2^{\text{desired}} & \quad P_3^{\text{desired}} & \quad \ldots & \quad P_M^{\text{desired}}
\end{align*}
\]

GUESS

\[
\begin{align*}
P_1 & \quad P_2 & \quad P_3 & \quad \ldots & \quad P_n
\end{align*}
\]

SCAN

\[
\begin{align*}
P_{1,\text{min}} & \quad P_{2,\text{min}} & \quad P_{3,\text{min}} & \quad \ldots & \quad P_{n,\text{min}} \\
\delta P_1 & \quad \delta P_2 & \quad \delta P_3 & \quad \ldots & \quad \delta P_n \\
P_{1,\text{max}} & \quad P_{2,\text{max}} & \quad P_{3,\text{max}} & \quad \ldots & \quad P_{n,\text{max}}
\end{align*}
\]

ACCURACY  I  R

COMPUTE
The "IDENT" card contains in columns 9-80 an identifying text and/or number, which is simply reproduced in the output for facilitating the filing of results.

The "PARAMETERS" card contains the integer \( n \), ending in column 20, signifying the number of parameters \( p_1, p_2, \ldots, p_n \) to be considered. The programme is limited to handle a maximum of 20 parameters.

The "CHECKPTS" card contains the integer \( M \), ending in column 20, signifying the number of checkpoints to be considered. The programme is limited to handle a maximum of 100 checkpoints.

The programme needs either a "DEGREE" card or a "POLYNOM" card. The "DEGREE" card contains the integer \( J \), ending in column 20, signifying the degree of the polynomial (2). All terms of degree \( J \) are then included, the polynomial ending with the term of \( J \)th power in \( p_n \). The number of terms \( N \) is in this case computed as the binomial coefficient

\[
N = \binom{n + J}{n}.
\]  \hspace{1cm} (13)

Alternatively, the "POLYNOM" card contains the integer \( N \), ending in column 20 directly. In this case \( N \) is not restricted to be one of the binomial coefficients (13). The programme is limited to handle a maximum of 100 terms.

The "VARIATIONS" card, containing only this codeword, proceeds the input of empirical data. Each parameter setting is now specified in the following manner: Starting on the next card the values of the \( n \) parameters \( p_1, p_2, \ldots, p_n \) are punched in fixed point decimal form \(^a\) using 8 number fields of 10 columns each per card. Then follows, using the same format \(^a\) and starting on a new

\(^a\) Fortran Format (6F10.0). This format permits omission of the decimal point for integers if these are punched ending to the extreme right in the number field of 10 columns.
card, the measured field values, \( E^m_{\text{measured}} \) (\( m = 1, 2, \ldots, M \)), at the
\( M \) checkpoints. The corresponding data for the next parameter
setting may now be inserted without each time repeating the
"VARIATIONS" card. However, it is also permitted to insert blocks
of such data, each headed by a "VARIATIONS" card. It should be
noted that the programme will not work unless the number of
parameter variations presented to it at least equals the number of
terms \( N \) specified by "DEGREE" or "POLYNOM".

The programme will remember parameter variations from a
preceding case, but will overwrite this information at the
appearance of the first "VARIATIONS" card. Thus, one cannot add
further empirical data to that of a previous case without reentering
all the original data.

The "DESIRE" card, containing only this codeword, precedes
the input of the desired field values, \( E^m_{\text{desired}} \) (\( m = 1, 2, \ldots, M \)), at
the \( M \) checkpoints. This data starts on the next card and uses the
same fixed point decimal format as above.

In order to start the iteration process of the Eqs. (11)
leading to the optimal parameter settings, one needs an initial set
of parameter values \( p_1, p_2, \ldots, p_n \). This data is preceded by either
a "GUESS" card or a "SCAN" card. The "GUESS" data should be used
whenever one for one reason or another has a reasonably good idea
regarding the whereabouts of the optimal solution. Following, the
"GUESS" card the \( n \) initial parameter values are specified directly
using the same fixed point decimal format as previously.

Without any such idea a completely wild guess might
easily lead to a false solution of the variational problem (7),
whereby a false minimum or even a saddlepoint or a maximum, may be
determined for the sum of squared differences. In such cases one
should better rely on the "SCAN" data, which follows the "SCAN" card
and consists of \( 3 \) sets of \( n \) numbers \( p_{k,\text{min}} \), \( 5p_k \), \( p_{k,\text{max}} \)
(\( k = 1, 2, \ldots, n \)), each set starting on a new card and using the same
fixed point decimal format as before. With this "SCAN" data available the programme will compute the sum of squared differences in Eq. (7) at each point in the n-dimensional mesh defined by the k\textsuperscript{th} parameter (k = 1, 2, \ldots, n) varying from \( p_{k, \text{min}} \) in steps of \( \delta p_k \) up to the largest value \( p_{k, \text{min}} + a\delta p_k \), (\( a = 1, 2, \ldots \)) which does not exceed \( p_{k, \text{max}} \). The point at which occurs the least sum is then taken as the initial point for the iteration process (11).

The iteration process is controlled by the "ACCURACY" card containing the integer I ending in column 20 and the fixed point decimal number R in columns 21-30. The object of both these quantities is to end the iteration loop. Thus, the integer I sets the maximum number of iterations to be tried. However, the iteration loop will also be stopped, if for each parameter the relative accuracies \( \left| \Delta p_k / p_k \right| \), (\( k = 1, 2, \ldots, n \)), does not exceed the number R. Here the \( p_k \) are the corrected parameters and the \( \Delta p_k \) the corrections resulting from the iteration (11).

It is permitted to set I = 0 or blank, in which case R is irrelevant and the iteration process (11) is not performed at all. The parameters resulting from the "GUESS" or "SCAN" data is then used directly in the final results (12). Thus it is possible to work the programme entirely by the "SCAN" process described earlier, although this is liable to consume considerably more computer time.

Another application of the case I = 0 is to compute suitable tolerances for the parameter settings. Having determined the optimal parameters, the effects of small errors may be studied at each checkpoint through an I = 0 case by inserting a "GUESS" with the optimal parameters slightly varied. The parameter tolerances may also be studied analytically by using the \( K \) checkpoint polynomials (2) or (3) and their first derivatives (9) which are available in the programme output.

As mentioned earlier the least squares polynomial fit by Eqs. (6) involves a smoothing of the empirical data read in under
"VARIATIONS", provided that the number of parameter variations exceeds the number N of terms taken in the polynomials. The results of this data smoothing may be studied by inserting one of the parameter variations as a "GUESS" in an $$I = 0$$ case, and comparing the computed fields to the measured fields at the M checkpoints. Conversely, one may in an $$I > 0$$ case insert these measured fields as a "DESIRE", and compare the computed parameters to the ones originally set.

Finally, the programme needs a "COMPUTE" card, containing only this codeword, which triggers a data checking process and initiates the computation.

In addition to the necessary data groups described above, the programme will also accept one or more cards headed by the codeword "REMARK". The rest of the card may then be used for remarks which will be reproduced during the listing of the data.

The data groups, headed by their codewords, may occur in any order with the following exceptions:

PARAMETERS must precede DEGREE, and GUESS or SCAN.
CHECKPTS must precede DESIRE.
VARIATIONS must be preceded by PARAMETERS, CHECKPTS, and DEGREE or POLYNOM.

The programme checks the input data for all possible formal data errors. Whenever an error is detected, this will be indicated in the programme output and the case skipped. It should in this connection be noted, that after the occurrence of an error the data from earlier cases no longer is preserved in the computer memory.
The following blocks of information are available as programme output:

1. Listing of input data for checking purposes.
2. "Polynomial Approximation" data, where for each of the \( M \) checkpoints the \( N \) coefficients of the polynomials (2) are listed in the order that they occur in this equation.

If, either intentionally or for reasons of badly chosen parameter variations, the system determinant of the linear Eqs. (6) vanishes, then one or more of the coefficients will be indeterminable. The programme handles these cases by setting such coefficients to zero, and gives a special notice about this in the output. To achieve this feature in spite of the unavoidable numerical truncation after 27 significant binary digits, it was necessary to compute the coefficient matrix of Eqs. (6) and perform its triangularization in double precision arithmetic. Furthermore, by properly interchanging rows in the course of the triangularizing process, it is arranged that the indeterminable coefficients occur as far to the right as possible in the polynomials (2). Consequently the \( n \) parameters should be presented in the order of their numerical significance, such that only the least significant terms are rejected and set to zero.

Finally, to gain experience about the computer time required, the time consumed up to the end of the polynomial approximation process is printed out.

The "polynomial approximation" data will not occur as output if this data is only preserved from a previous case.

3. "Scanning Result". If a scanning process, initiated by the "SCAN" data, is involved in the computation, the \( n \)-dimensional meshpoint at which the sum of squared residues in Eq. (7) is smallest, is indicated in the output.
In addition, the time consumed during the scanning process is printed out. The scanning mesh must not be chosen too fine in cases involving many parameters in order to avoid excessive scanning times.

4. "Convergence Check". This output is provided in order that one may follow the iteration process (11). Here the initial values $p_i$, the corrections $\Delta p_i$, the final values $p_i + \Delta p_i$, and the relative accuracies $|\Delta p_i/p_i|$ (i = 1, 2, ..., n) are given after each iteration. After the iteration process is completed the programme inspects the second partial derivatives of the sum of squared residues in Eq. (7) to determine whether or not a true minimum sum, and not only a saddle or maximum, has been obtained, and makes a note of the result in the output. Finally the time consumed during the iteration process is printed out.

5. "Final Results". Here the optimal settings of the n parameters are given as well as the desired result, the final result, and the residues at each of the M checkpoints.

6. "Partial Derivatives at Optimum". The first order partial derivatives of the final resultant fields with respect to the n parameters are listed for each of the M checkpoints. The object of this data is to provide useful information for estimating the parameter tolerances.

Finally the total computer time used for the case is printed out.

OTHER APPLICATIONS

The previous description is presented in terms of shimming and trimming directly the field of a magnet. However, other applications are also possible.

One might for instance consider a case where the equivalent length of a magnetic lens is to be corrected such that
it becomes constant over the aperture. The optimal correcting elements may then be determined by the programme if one for the empirical results $B^m$ inserts line integrals $\int B^m ds$ through the lens. These may be either measured directly or calculated separately from field measurements. The "checkpoints" will in this case be a chosen set of integrating paths.

It is obvious that one might in general consider using the programme to correct any type of measurable field depending on a set of controllable parameters, such that the field or some function involving the field attains desired values or properties.