7

MASTER

SINGLE PASS COLLIDER MEMO CN-342

AUTHOR: Chris Hawkes[†] and Martin Lee
DATE: September 2, 1986
TITLE: Recent Upgrading of the Modelling Program COMFORT*

1. Introduction

The computer code COMFORT, developed for the online control of machine functions at the SLC, has recently undergone several modifications to overcome some of its limitations. This note describes the reasons for these changes, the methods employed, some test results and the applications of the new version of the program.

SLAC-CN--342

DE86 015621

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

* Work supported by the Department of Energy, contract DE-AC03-76SF00515

† California Institute of Technology, Pasadena, California 91125;
supported by Department of Energy, contract DE-AC03-81-ER40050

2. The COMFORT Process

The present version of COMFORT, called COMFORT3[1-4], is used for online lattice configuration changes in the control program for SLC systems, such as the LINAC and Damping Rings, as well as for SPEAR and PEP. For the SLC systems this is referred to as the COMFORT process. The lattice configuration for each system is specified by a set of lattice functions (the Twiss parameters) at each of several specific fit points along the beam line. The COMFORT process consists of:-

- (a) the user inputs the values of the lattice functions to be changed;
- (b) COMFORT computes the strengths of lattice elements (e.g. quadrupole magnets) needed to produce the desired changes.

For example [2], the configuration of a Damping Ring is defined by the horizontal and vertical tunes (ν_x, ν_y), and the periodicity conditions on the lattice functions (α, β, η) across a cell. The fit points are taken to be the beginning and midpoint of a cell. The operator can change the values of the tunes of the Damping Ring by entering the desired values. COMFORT computes the strengths of five sets of quadrupole magnets needed to give these tune values, while still maintaining the overall periodicity conditions in a cell. This particular computation involves solving numerically a set of five non-linear equations specifying the lattice functions at the fit points, with five unknowns - the quadrupole strengths. In general, the numbers of equations and unknowns depends on the system.

3. The Computation

COMFORT solves the problem numerically using a fitting algorithm which minimises the sum of squares:

$$\sum_i W_i \{G_i - F_i(K)\}^2 \quad (1)$$

where

G_i = the desired value of a lattice function at a fit point

F_i = the value of the lattice function at the fit point
for a given K

W_i = a weighting factor

$K \equiv k_1, \dots, k_j$ = the variables (e.g. quadrupole strengths)

For the case of the Damping Ring described in Section 2, F_1 and F_2 can be the horizontal and vertical tunes, F_3 can be the difference in α_x between the beginning and end of a cell, etc.; G_1 and G_2 are the desired tunes input by the user, $G_3 = 0$, etc., to impose the correct periodicity conditions. The program varies $k_1, \dots, k_j$ from their initial values in order to minimise the sum of squares (1). The solution is then the set of quadrupole strengths to give the best fit to the desired values. If the sum (1) becomes zero, this represents an exact solution to the problem.

4. The Limitations

In COMFORT3 the minimisation procedure described in Section 3 is performed using the least squares fitting program LMDIF [5]. In practice, some of the solutions found by COMFORT3 cannot be used because the fitted K represents magnet strengths beyond the bounds imposed by the operating range of the magnet power supplies. This is because LMDIF, as well as fitting programs used in earlier versions of COMFORT, does not allow the user to input bounds on the variables K .

Another limitation is that the only way to relax the constraints of the fit, in cases where exact or useful solutions cannot be found, is by assigning lower weighting factors, W_i , to the less crucial lattice functions. In practice, this facility has proved difficult to use effectively.

5. The First Modification Using NPSOL

In order to overcome the limitations outlined in section 4, LMDIF has been replaced with a new fitting program called NPSOL [6]. This preliminary modification to the COMFORT3 program will be referred to as COMFORT_SOL. NPSOL has the following features:

- (a) It can minimise general non-linear functions of the variables, K , and not just a sum of squares.
- (b) It allows the user to impose upper and lower bounds on each variable.
- (c) It allows the user to impose upper and lower bounds on any general non-linear function of the variables.

As an initial test of COMFORT_SOL, its performance was compared to that of COMFORT3, which has already been used extensively in many situations. Of course, this meant that only standard fitting problems could be used, and the new upper and lower bound facilities, which COMFORT3 does not accommodate, were not being tested. COMFORT_SOL consistently required several times more CPUtime than COMFORT3 to solve a typical problem. In extreme cases, COMFORT_SOL took up to ten times as long, and then found a different solution to that found by COMFORT3. Some simple rearrangement of the code and optimisation of the NPSOL control parameters[†] was found to speed up COMFORT_SOL by a factor of about three, depending on the problem.

Occasionally, COMFORT_SOL would fail to find any solution to a problem which COMFORT3 could solve successfully. This turned out to be caused by the way in which the physical bounds of ± 2 on the traces of the ring transport matrices in a storage ring configuration are handled inside COMFORT. The method used with LMDIF in COMFORT3 was incompatible with NPSOL in COMFORT_SOL. Consequently, a further modification to COMFORT_SOL was made to input these physical upper and lower bounds into NPSOL itself, using feature (c) above, the traces of the transport matrices being non-linear functions of the lattice variables. This improved the performance and stability of COMFORT_SOL.

[†] In our test runs we have found the performance of COMFORT to depend critically upon the values of the control parameters input to NPSOL. The values provided with COMFORT4 as default in NPSLAC.OPTIONS are those which have been found to be most effective with the COMFORT_SLAC version of the program to be described in Section 6. They work well if the fitting process is started from a point not too far away from the solution, as will be typical for online applications. For more difficult problems, users may find it useful to increase the values of the parameters called "Function Precision", "Feasibility Tolerance" and "Optimality Tolerance". We also advise consultation of sections 5, 6 and 7 of reference 6.

6. The Second Modification Using NPSLAC

After optimisation, COMFORT_SOL still takes significantly longer than COMFORT3 to solve the same problem. To speed it up further it is necessary to modify NPSOL.

One time-consuming feature of COMFORT_SOL is that for each iteration of the variables **K** during the fit, the complete lattice calculation has to be performed twice. This is because of the way in which NPSOL calls two user-supplied subroutines (i.e. supplied by COMFORT_SOL) [7]:

OBJFUN – to calculate the sum of squares (1)

CONFUN – to calculate the bounded non-linear functions

At our request, the authors of NPSOL have now modified their code so that only one evaluation is necessary for each iteration, thus halving the time taken by COMFORT_SOL. This modification to NPSOL has now been incorporated as a standard feature in the latest version of the NPSOL program [8].

Another saving in time has been achieved by making NPSOL more specialised. As mentioned in section 5, NPSOL can minimise general non-linear functions, whereas in the standard uses of COMFORT, we need only to minimise a sum of squares. (Each element of the sum of squares is a non-linear function of **K**.) Some of the overhead involved in running NPSOL can be avoided in this case. Hence, we requested the authors of NPSOL to provide us with a special version which handles only least squares problems. This version is called NPSLAC [9] and the version of COMFORT using NPSLAC will be referred to as COMFORT_SLAC. After re-optimising the control parameters, COMFORT_SLAC was tested on the same problems as was COMFORT_SOL and was found to perform satisfactorily. Typically COMFORT_SLAC takes about 50% more CPUtime than COMFORT3 requires for the same problem. In addition, the upper and lower bound facilities of COMFORT_SLAC (see section 7) have been successfully tested.

COMFORT_SLAC has now been made the latest official version of COMFORT, called COMFORT4.

7. Using the New Features of COMFORT4

For the general user, the important new features of COMFORT4 are the upper and lower bound facilities. The use of these is illustrated with the following examples of COMFORT4 input control statements:

```
QFH : QUAD, L=0.0828, K1=-18.32  
VARY, QFH , K1, MIN=-20., MAX=20.
```

In most applications, the upper and lower bounds on the variables should be set to correspond to the practical limits to the range of the magnet power supplies.

```
FUNCTION, MIDCELL, AX, MIN=-0.01, MAX=0.01  
FUNCTION, MIDCELL, AY, MIN=-0.01, MAX=0.01
```

The upper and lower bounds on the lattice functions can be used to relax the constraints on less critical functions in situations where an exact fit is not required. A new command has to be used for each function. It should be noted, however, that in the present version of NPSLAC there must be at least one element in the sum of squares (1), and hence the user must leave at least one lattice function as a true fit point, rather than give it an upper and lower bound.

8. Summary

Many users have been satisfied with the performance of COMFORT3, but some have found it to have unnecessary limitations. COMFORT4 overcomes some of the limitations of COMFORT3.

Some users have tried early versions of COMFORT4 (COMFORT.SOL) and have not been satisfied. The latest version of COMFORT4 (COMFORT.SLAC) has the same reliability as COMFORT3 and takes only slightly longer.

COMFORT4 is available to supersede COMFORT3.

Acknowledgements

Much of the early work of implementing NPSOL in COMFORT was done by Lou Sanchez.

We are very grateful to Margaret Wright and her collaborators at the Systems Optimization Laboratory of Stanford University for providing us with special versions of their fitting programs (NPSOL and NPSLAC), and for her interest and patience when dealing with our problems.

References

1. M.D. Woodley, M.J. Lee, J. Jäger, A.S. King: SLAC-PUB-3086 (March 1983) published in IEEE Trans. Nucl. Sci., NS-30: no. 4, 2320 (August 1983)
2. M.J. Lee, J.C. Sheppard, M. Sullenberger, M.D. Woodley: SLAC-PUB-3217 (Sept 1983)
3. I. Almog, J. Jäger, M. Lee, M. Woodley: SLAC-PUB-3608 (March 1985)
4. L. Sanchez, H. Shoaee, M. Woodley: COMFORT Version 3.0 Writeup (in preparation)
5. Jorge J. More, Burton S. Garbow, Kenneth E. Hillstom: User Guide for MINPACK-1, ANL-80-74 (Argonne, August 1980)
6. Philip E. Gill, Walter Murray, Michael A. Saunders, Margaret H. Wright: User's Guide for NPSOL (Version 4.0), SOL 86-2 (Stanford, January 1986)
7. See section 4 of reference 6.
8. Philip E. Gill, Walter Murray, Michael A. Saunders, Margaret H. Wright: NPSOL (Version 4.02) (July 1986)
9. Margaret H. Wright: private communication, NPSLAC (Version 4.02) (July 1986)