

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN – SL DIVISION

DFS User Guide V1.0

The LEP Dispersion Free Steering Program User Guide

J. Wenninger

Abstract

The purpose of this note is to explain the options of the LEP Dispersion Free Steering program. Two examples are also explained.

This guide is valid for DFS program versions 3.0 and 3.1

Geneva, Switzerland

June 21, 2001

1 Introduction

The purpose of this note is to give a brief description of the Dispersion Free Steering (DFS) program User Interface (UI). The UI main window is shown in Fig. 1. A DataViewer (DV) is running as a separate process. Interaction between UI and DV is over UNIX signals. Please refer to the DV user guide for details.

The UI is designed to handle up to 2 different beams at the same time. It is possible to work on any of the 2 beams, the average of the 2 beams or on both beams at the same time. It is therefore possible to provide different response matrices for the 2 beams, which is useful for machine with separate rings or with very large energy sawtooth like LEP. The program takes care of rebuilding the response matrices based on the user input (see below). Note that when 2 beams are handled at the same time, the size of the response matrix is doubled.

The orbit and dispersion can be corrected using both SVD and MICADO correction algorithms.

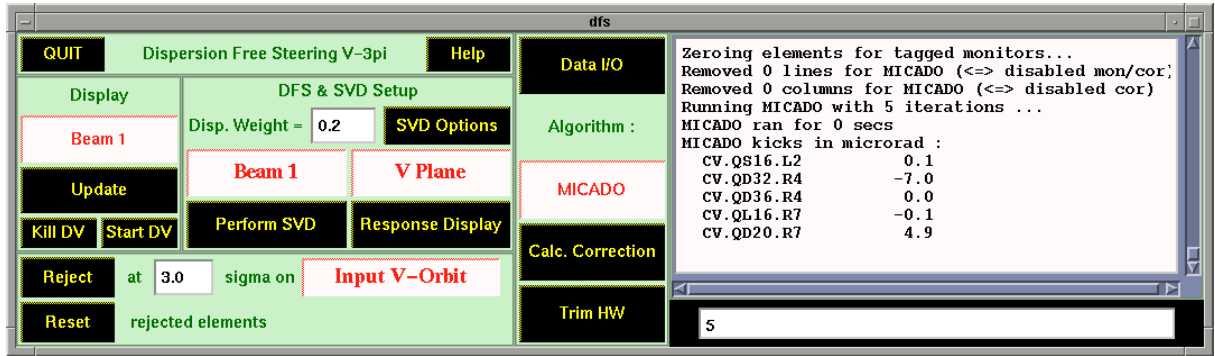


Figure 1: The main UI window of the DFS program. Informations are printed on the scrolled text on the right side.

1.1 Implementation of SVD

The SVD algorithm [1] is a powerful tool to handle singular systems and to solve them in the least square sense. The classical decomposition of matrix \mathbf{T} with dimension $N \times M$, $N \geq M$, has the form

$$\mathbf{T} = \mathbf{U}\mathbf{W}\mathbf{V}^t = \mathbf{U} \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & & \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & w_M \end{pmatrix} \mathbf{V}^t, \quad (1)$$

where \mathbf{W} is a diagonal $M \times M$ matrix with non-negative diagonal elements. \mathbf{V}^t is the transpose of the $M \times M$ orthogonal matrix \mathbf{V} ,

$$\mathbf{V}\mathbf{V}^t = \mathbf{V}^t\mathbf{V} = \mathbf{I}, \quad (2)$$

while \mathbf{U} is an $N \times M$ column-orthogonal matrix

$$\mathbf{U}^t\mathbf{U} = \mathbf{I}. \quad (3)$$

The vector $\vec{\vartheta}^{(i)}$, corresponding to the i^{th} column of matrix \mathbf{V} ,

$$\vec{\vartheta}^{(i)} = \begin{pmatrix} V_{1i} \\ V_{2i} \\ \dots \\ V_{Mi} \end{pmatrix}, \quad (4)$$

is also an eigenvector with eigenvalue $w_i^2 \geq 0$ of the $M \times M$ symmetric matrix $\mathbf{T}^t\mathbf{T}$ [2]. This property is used in the DFS program where matrix \mathbf{T} is first multiplied with its transpose \mathbf{T}^t . An eigenvalue search is then applied on the resulting symmetric matrix $\mathbf{T}^t\mathbf{T}$. This eigenvalue decomposition can be performed with a variety of algorithms, including SVD. In the DFS program three eigenvalue search algorithms have been implemented to check the numerical stability, in particular for very large matrices (up to dimension $\approx 2300 \times 300$).

The orbit and dispersion response $\vec{v}^{(i)}$ associated to kick eigenvector $\vec{\vartheta}^{(i)}$

$$\vec{v}^{(i)} = \mathbf{T}\vec{\vartheta}^{(i)} = \mathbf{U}\mathbf{W}\mathbf{V}^t\vec{\vartheta}^{(i)} = w_i \begin{pmatrix} U_{1i} \\ U_{2i} \\ \dots \\ U_{Ni} \end{pmatrix} \quad (5)$$

corresponds (modulo a factor w_i) to the i^{th} column of matrix \mathbf{U} . This vector is easily reconstructed once the $\vec{\vartheta}^{(i)}$'s are known.

Provided that all w_i are nonzero, the SVD decomposition can be used to obtain a *pseudo-inverse* of matrix \mathbf{T} ,

$$\mathbf{V}\mathbf{W}^{-1}\mathbf{U}^t. \quad (6)$$

Singular elements, corresponding to values of w_i that are very small or zero, can be regularized by setting $1/w_i$ to 0 in matrix \mathbf{W}^{-1} .

The matrix \mathbf{T} is defined as

$$\mathbf{T} = \begin{pmatrix} (1 - \alpha)\mathbf{A} \\ \alpha\mathbf{B} \\ \beta\mathbf{I} \end{pmatrix}, \quad (7)$$

where response matrix \mathbf{A} (dimension $N \times M$) is used to describe the relation between corrector kicks and beam position changes at the monitors. The element A_{ij} of the response matrix corresponds to the orbit shift at the i^{th} monitor due to a unit kick from the j^{th} corrector. \mathbf{B} is the $N \times M$ dispersion response matrix, its elements B_{ij} giving the dispersion change at the i^{th} monitor due to a unit kick from the j^{th} corrector. The weight factor α is used to shift from a pure orbit ($\alpha = 0$) to a pure dispersion correction ($\alpha = 1$). β is a kick weight and \mathbf{I} is the identity matrix. For more details refer to Reference [2].

2 Display Options

The options for the display are located in the area labelled *Display* on the left side of the UI. It is possible to display the data for either *Beam 1*, *Beam 2*, *<Beam 1 & Beam 2>* (average of the 2 beams) or *Beam 1 - 2* (the difference between beam 1 and 2). Clicking on **Update** refreshes the display with the requested data.

In case of problems, it is possible to kill or restart the DV process with the buttons **Kill DV** and **Start DV**.

3 DFS & SVD Setup

The parameters for DFS and SVD are available in the central part of the UI.

The main parameter for DFS is the weight factor for the dispersion, which by default is set to *Disp. Weight = 0.2*. For a pure orbit correction the weight can be set to 0. More advanced options are available by clicking on **SVD Options**, which will pop up the window of Figure 2. The options include :

- The choice of 3 algorithms for eigenvalue search : *Singular Value Decomposition* (SVD, the default), *Jacobi Rotation* and *QL Decomposition* [1]. SVD and QL decomposition are the fastest algorithms. The Jacobi Rotation algorithm is more robust (although a factor ≈ 4 slower than SVD) and is well suited for very singular situations or very large matrices.
- The numerical precision used inside the algorithms to stop some internal convergence loops. The default is set to 10^{-10} .
- The cut on the eigenvalues (normalised to the largest one). Eigenvalues that are smaller than this cut are set to zero and ignored for corrections to avoid singularities.
- A bias on the momentum offset.
- A weight for the corrector kicks. This corresponds to the parameter β in Reference [2]. By default $\beta = 0.1$, but its value must be adapted for each machine depending on the typical kick strength.
- For the MICADO algorithm, the constrains on the corrector kicks can be applied on the kick *increments* (default) or on the *absolute* kick value. **Note that the later option may not work properly !!**.

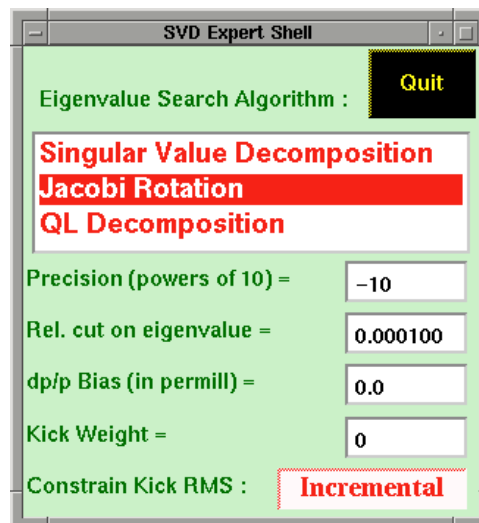


Figure 2: The DFS expert option window.

The correction algorithm can be applied to *Beam 1* , *Beam 2* , *<Beam 1 & Beam 2>* (the average of the 2 beams) or *Beam 1 & 2* (both beams simultaneously, with independent response matrices for each beam). The plane can be selected to be *Horizontal* or *Vertical* .

For a correction of the horizontal dispersion, the reference dispersion is first subtracted from the measured dispersion !

To run the SVD decomposition (after setting up the right options (beam, plane, ...)) the user must click on **Perform SVD**.

The button **Response Display** pops up the window shown in Fig. 3. This window allows to display :

- The orbit, dispersion ($\vec{v}^{(i)}$) and corrector kick eigenvectors ($\vec{\vartheta}^{(i)}$) corresponding to the selected eigenvalue i (in descending order of importance).
- The orbit and dispersion response for a selected corrector magnet (matrix \mathbf{T}).
- The inverse response of a selected monitor (the pseudo-inverse of \mathbf{T}).

In each case the user is requested to enter the element number (*starting from 0 !!*). Clicking on the associated **Display** button will automatically fill the display with the selected data.

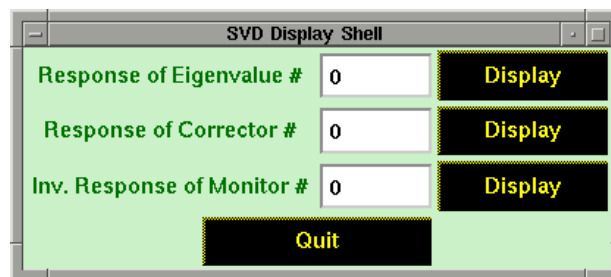


Figure 3: Option window to select the display of the various responses.

4 Element rejection

The lower part of the UI allows to reject (i.e. disable) elements that are beyond a selected number of standard deviations from the mean value. This is useful to automatically reject elements that given wrong or suspicious readings. The rejection can be applied to the orbit and dispersion of each plane or on the orbit and dispersion results. The correction can then be iterated...

It is possible to *disable*, *enable* or *zero* the elements of a given region by clicking on the **Select** button of the DV (top right). After selection of a given action, the user is requested to click on the display and choose the area on which the action has to be applied. Please note that this option works only on the DV Views *Orbit + Disp H* and *Orbit + Disp V* (see **Views** selection, top right of DV window).

5 Data IO

Data can be input by clicking on **Data I/O** in the UI, which will pop up the window shown in Fig. 4. The top part of the window deals with the orbit and kick data input. There are 4 data input formats :

- A standard LEP orbit format (binary).
- A standard LEP dispersion format (binary).
- A MAD [3] orbit and dispersion input format (see also the examples).
- A MAD kick input format.

A C-programmer can easily extend (or replace items in) this list to other data formats.

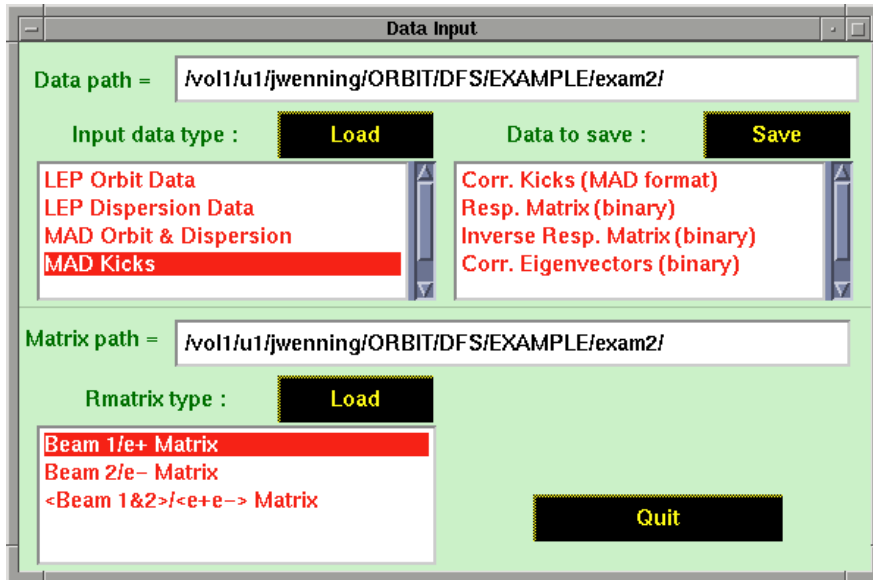


Figure 4: The data input/output window.

It is possible to save the corrector settings predicted for a correction in a MAD [3] compatible format. The various response or eigenvalue matrices can be stored in a binary format. Again, those options can be easily extended or replaced.

The lower part of the window allows to define the response matrices for *Beam 1* , *Beam 2* , *<Beam 1 & Beam 2>* (the average of the 2 beams). The file format must be build using/adapting the appropriate programs (see examples).

6 DFS Corrections

Four correction algorithms/types are available :

- *SVD*
- *SVD on Bare*
- *MICADO*
- *MICADO on Bare*

For **Bare** corrections, the effect of all correctors is first unfolded. The correction is evaluated on the unfolded orbit and dispersion [2]. This effectively corresponds to re-seeded completely all the corrector settings. It also allows to obtain the raw orbit and dispersion in the absence of any

correction (if the user requests 0 eigenvalues/correctors used for the correction). Note however that due to non-linearities (i.e. sextupoles) this unfolded data does not necessarily correspond to the true raw orbit and dispersion.

Once the correction type is selected, clicking on **Correction** will run the algorithm on the data. The user is prompted for the number of correctors (MICADO) or eigenvalues (SVD). Before evaluating a correction with SVD, the SVD decomposition of the matrices must first be performed (**Perform SVD**). After a successful correction the display will be updated with a view of the orbit, dispersion and corrector kick r.m.s. as a function of the number of correctors (MICADO) or number of eigenvalues (SVD), see Fig. 5. This plot can be used to optimise the number of correction elements.

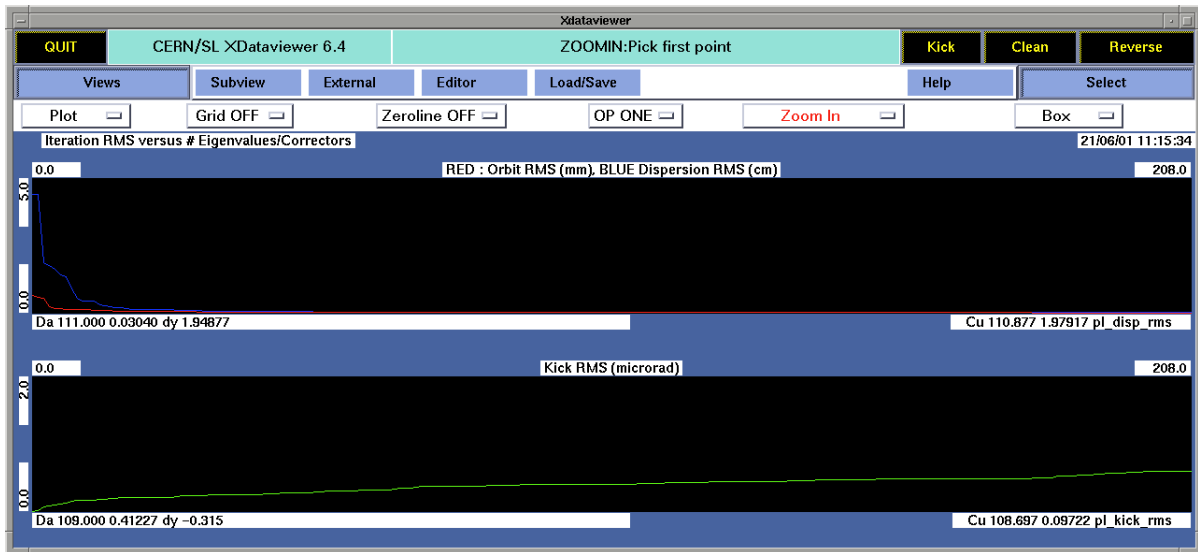


Figure 5: Example of the display of the orbit, dispersion (top) and corrector kick (bottom) r.m.s. as a function of the number of eigenvalues.

A correction can be send to the machine over the **Trim HW** button. This however requires that the C-function **dfs_trim_tohw** in file **dfs_trimhw.c** be replaced by proper code.

7 Example 1

This example is based on the LEP machine. It is a test for a simple orbit correction (no dispersion is involved for this example). The input data for the DFS program is generated using the MAD program [3] command file **twiss.mad**. This command file generates four ASCII output files :

- cor.optics.out : contains the twiss parameters for the LEP orbit corrector magnets.
- mon.opics.out : contains the twiss parameters for the LEP monitors.
- cor.set.dfs : contains the corrector kicks (all set to zero).
- mon.set.dfs : contains the orbit and dispersion at each monitor.

The files with extension **.out** are used by the program **twiss_to_matrix** of the DFS package to build the orbit response from the twiss parameters (betatron functions and phases) based on the classical formulae. Type **twiss_to_matrix twiss.in** to build the response matrix file **resp.matrix** which has the right format for the DFS program.

Once the response file **resp.matrix** is generated, start the DFS program from this example path with **dfs -cD -cR**. With the arguments **-cD** and **-cR** the file path for the input data and response matrices will be set to the current directory. Then :

- Select **Beam 1** for both display and DFS setup.
- Click on **Data IO** .
- Select **MAD Orbit and Dispersion** as Input Data Type.
- Click on **Load** and select the file **mon.set.dfs**.
- Select **MAD Kicks** as Input Data Type.
- Click on **Load** and select the file **cor.set.dfs**.
- Select **Beam 1/e+ Matrix** as Rmatrix Type.
- Click on **Load** and select the file **resp.matrix**.
- Set the weight for the dispersion to **Disp Weight = 0**.

The input is setup. It is now possible to search for eigenvalues by clicking on **Perform SVD** . After running SVD the display should be updated with the eigenvalue spectrum as shown in Fig. 6. Options can be changed as explained before in this document.

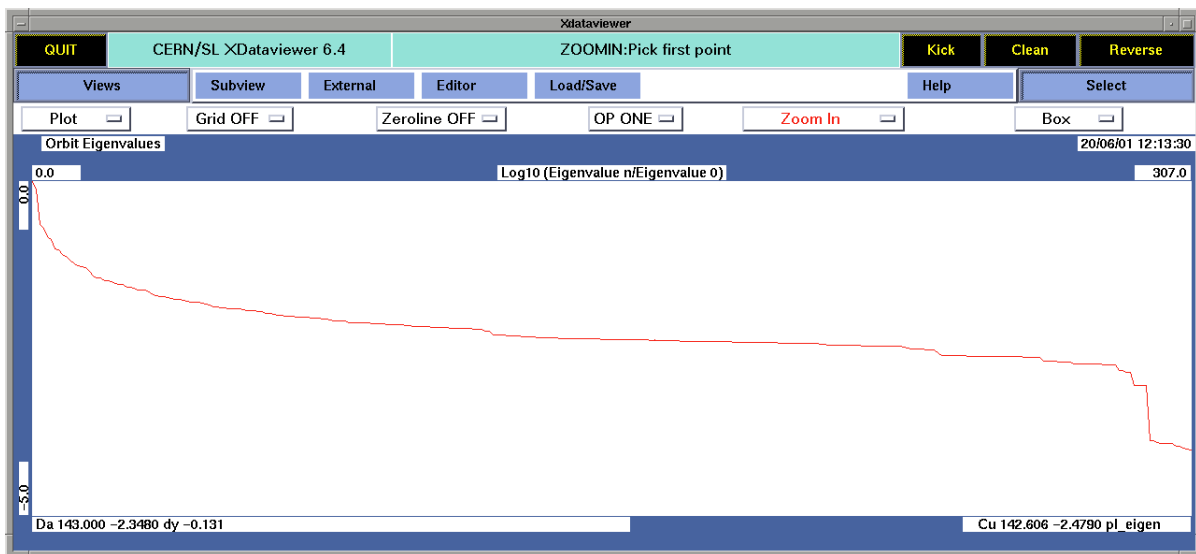


Figure 6: Eigenvalue spectrum obtained with a kick weight of 0 and the Jacobi Rotation algorithm for example 1.

8 Example 2

This examples performs a full DFS correction for LEP.

The shell script **run.corrector.loop** is first used to compute the orbit and dispersion response for a selected sample of vertical orbit corrector magnets. Each corrector is set in turn to a strength of $5 \mu\text{rad}$ and 2 orbits are computed for different momentum offsets. The orbits for corrector No. **i** are stored in path **tmp** with names “**respi.out1.gz**” and “**respi.out2.gz**” (in zipped format to save space). In total there are $2 * 208$ such response files and 2 files with the twiss parameters (betatron function and phase, nominal dispersion).

The orbit and dispersion response is computed from the output files using program **make_matrix** of the DFS package. First the files in path **tmp** must be un-zipped. Running the command **make_matrix matrix.in** will generate a file **resp.matrix** which contains the orbit and dispersion response.

As a test case the MAD command file generates a perturbed closed orbit and dispersion using 2 corrector magnets (CV.QD32.R4 set to $7.0 \mu\text{rad}$, CV.QD20.R7 set to $-5.0 \mu\text{rad}$). The script generates 2 input files for the DFS program, **mon.set.dfs** and **cor.set.dfs**.

To run this example first start the DFS program. Start the program from this example path with **dfs -cD -cR**. With the arguments **-cD** and **-cR** the file path for the input data and response matrices will be set to the current directory. Then :

- Select *Beam 1* for both display and DFS setup.
- Click on **Data IO** .
- Select *MAD Orbit and Dispersion* as Input Data Type.
- Click on **Load** and select the file **mon.set.dfs**.
- Select *MAD Kicks* as Input Data Type.
- Click on **Load** and select the file **cor.set.dfs**.
- Select *Beam 1/e+ Matrix* as Rmatrix Type.
- Click on **Load** and select the file **resp.matrix**.

The input is then setup. The DV display should correspond to Fig. 7. It is now possible to search for the response matrix eigenvalues by clicking on **Perform SVD** . After running SVD the display should be updated with the eigenvalue spectrum as shown in Fig. 8. Options can be changed as explained before in this document.

To test the program with this example it is best to use the algorithm :

- Set the algorithm to *MICADO* .
- Click on **Calc. Correction** .
- When you are prompted for the number of correctors, enter **5**.

As a results the program should indicate a list of proposed kicks. Two of them should be :

- CV.QD32.R4 -7.0
- CV.QD20.R7 4.9

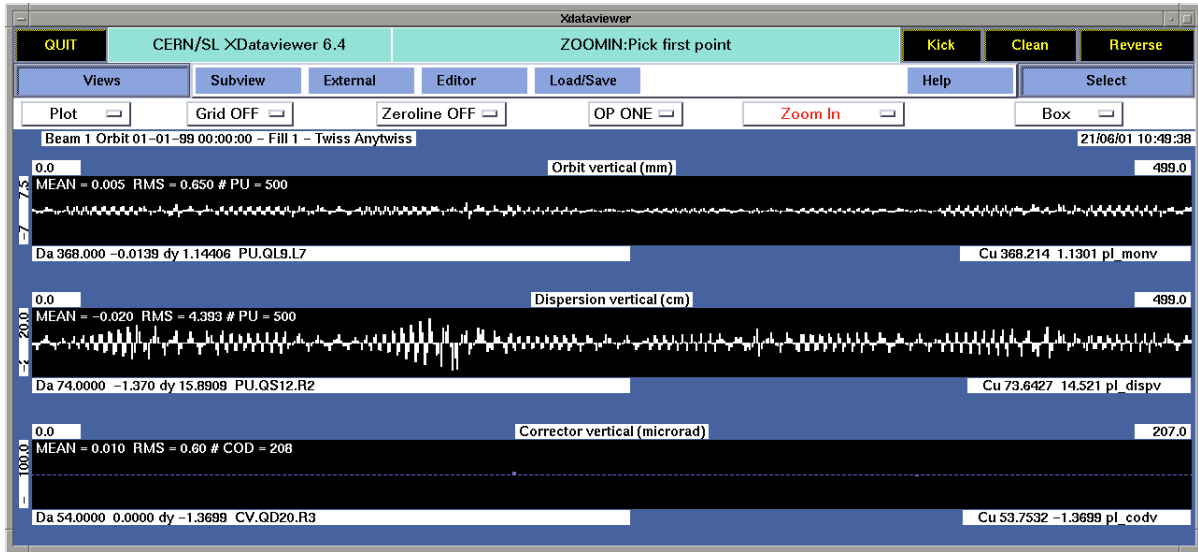


Figure 7: Input data for example 2 (vertical plane). The orbit (top), dispersion (centre) and corrector kicks (bottom) are indicated.

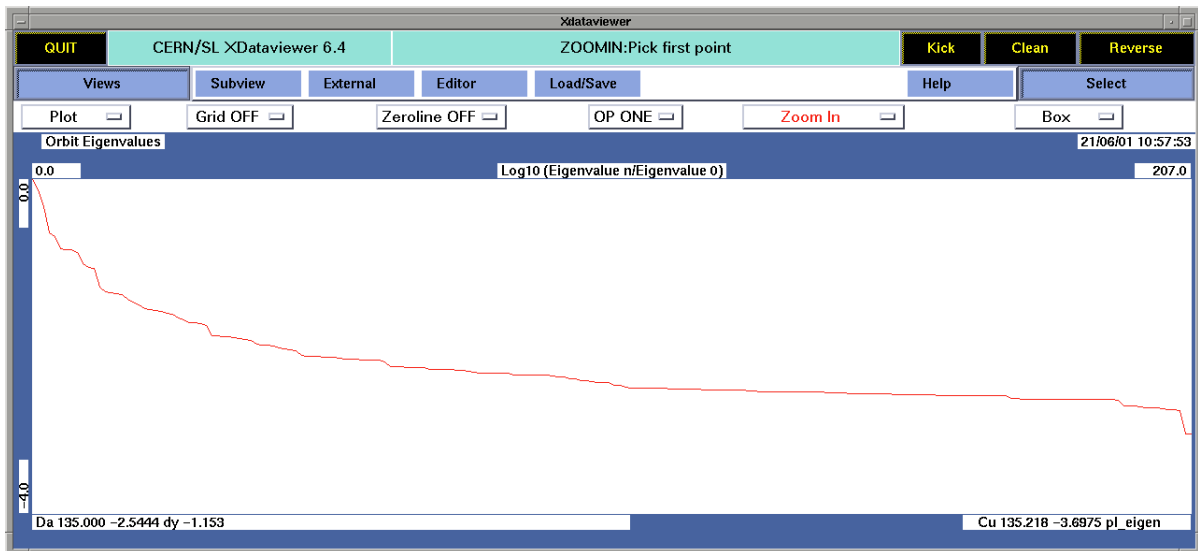


Figure 8: Eigenvalues for example 2. The dispersion weight is set to 0.2 and the kick weight to 0 (expert option).

This is exactly the opposite of the kicks that were used to generate the orbit and dispersion perturbation. From the **Views** selection of the DV, it is possible to view the *Orbit Results* shown in Fig 9 and the *Dispersion Results* shown in Fig 10. Note that neither the orbit nor the dispersion r.m.s. are exactly 0 after the correction. This is coming from non-linearities due to the LEP sextupoles which introduce a weak dependence of the response matrix on the closed orbit amplitude.

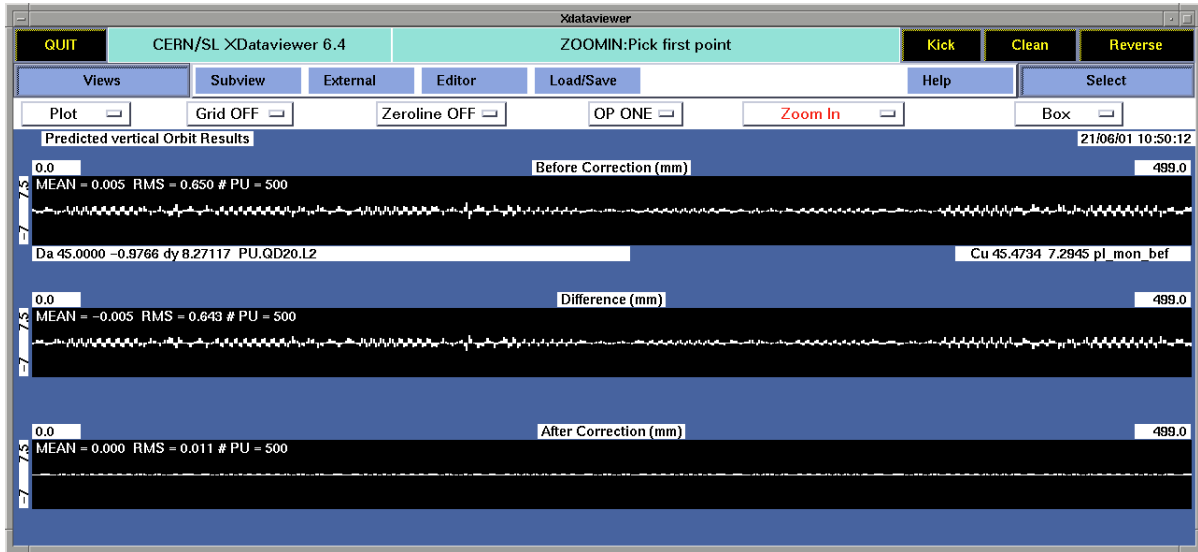


Figure 9: Correction result for the (vertical) orbit of example 2 using the MICADO algorithm with 5 correctors.

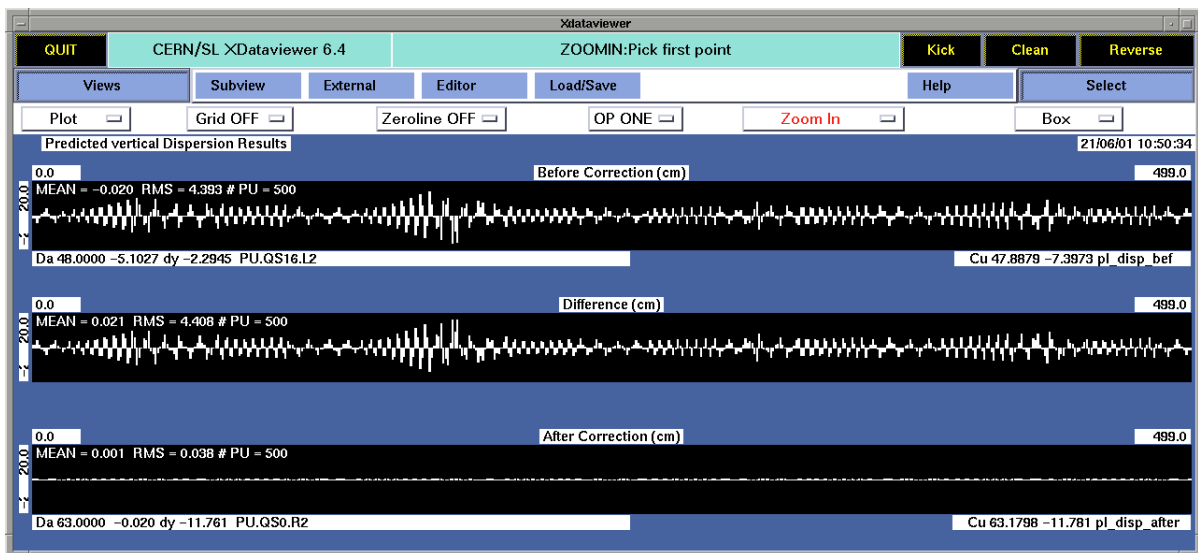


Figure 10: Correction result for the (vertical) dispersion of example 2 using the MICADO algorithm with 5 correctors.

9 Appendix A : Units

The programs delivered with the DFS package (`dfs`, `make_matrix`, `twiss_to_matrix`) assume that :

- Closed orbits are always in **mm** (MAD convention).
- The reference or input dispersion is given in **m** (MAD convention).

- The **dfs** programs assumes that :
 - The orbit response is given in **mm/μrad**.
 - The dispersion response is given in **m/μrad**.
 - The input kicks are given in **rad** (MAD convention).
 - Internally the units are **mm** for the orbit, **cm** for the dispersion and **μrad** for the corrector kicks.

10 Appendix B : Response Matrix Input Formats

In order to handle large matrices more conveniently, the response matrices are stored in binary format files. Each file is first subdivided in a *HEADER*, *VECTOR DATA* and *MATRIX DATA* section (in this order !).

10.1 Header Data

Header information is in pure ASCII and has the form :

```
#DFS-MATRIX v DATA
#MACHINE v LEP
#TWISS v g05150b99v5
#PLANE v vertical
#MTYPE v ORBIT
#COMMENT v This is example No. 2
#ENERGY v 100.000
#QH v 98.2841
#QV v 96.1960
#NUM-MON-H v 500
#NUM-MON-V v 500
#NUM-COR-H v 175
#NUM-COR-V v 208
```

The meaning of the keywords is :

- MACHINE : the machine name
- TWISS : the twiss/optics file name
- PLANE : the plane (VERTICAL or HORIZONTAL)
- MTYPE : the matrix type (ORBIT or TRAJECTORY)
- COMMENT : a comment...
- ENERGY : the beam energy
- QH : the horizontal tune
- QV : the vertical tune
- NUM-MON-H : the number of hor. monitors
- NUM-MON-V : the number of vert. monitors
- NUM-COR-H : the number of hor. correctors
- NUM-COR-V : the number of vert. correctors

10.2 Vector Data

The vector type data is subdivided into STRING type data (elements names) and FLOAT type data (element betatron functions...). The STRING type data blocks starts with the keyword :

#NAME-VECTOR v DATA

Each data block begins with :

#DATA v keyword

#STR-VECTOR s number

where **number** is the number of elements and **keyword** indicates the type of data. Valid keys are :

- MON-NAME-H : horizontal monitor names
- COR-NAME-H : horizontal corrector names
- MON-NAME-V : vertical monitor names
- COR-NAME-V : vertical corrector names

The data block keyword is followed by **number** lines holding each one element name.

The FLOAT type data starts with the keyword :

#DATA-VECTOR v DATA

followed by data blocks. Each data block begins with :

#DATA v keyword

#DATA-VECTOR s number

where **number** is the number of elements and **keyword** indicates the type of data. Valid keys are :

- MON-BETA-H : the hor. betatron function at the hor. monitors.
- MON-MU-H : the hor. betatron phase at the hor. monitors.
- MON-DISP-H : the hor. dispersion at the hor. monitors (DFS output).
- MON-ORB-H : the hor. orbit at the hor. monitors (DFS output).
- MON-BETA-V : the vert. betatron function at the vert. monitors.
- MON-MU-V : the vert. betatron phase at the vert. monitors.
- MON-DISP-V : the vert. dispersion at the vert. monitors (DFS output).
- MON-ORB-V : the vert. orbit at the vert. monitors (DFS output).
- EIGENVALUE : the eigenvalues of the response matrix (DFS output).

Items tagged with *DFS output* are not used as input for the response files, but as output from the dfs program (for further use...). The two keyword lines are followed by a (binary) block of floats with length equal to **number**.

10.3 Matrix Data

The matrix type data blocks start with the keyword :

```
#DATA-MATRIX v DATA
```

Each individual matrix data block begins with :

```
#DATA v keyword  
#MATRIX s ncol x nrow
```

where **ncol** is the number of columns, **nrow** the number of rows and **keyword** indicates the type of matrix. Valid keys are :

- RESP-ORB : the orbit response matrix **A** in **mm/mrad**
- RESP-DISP : the dispersion response matrix **B** in **m/mrad**
- RESP-INV : the pseudo-inverse of the response matrix **T** which combines the orbit and dispersion responses with the proper weights (DFS output)
- RESP-MAT : the response matrix **T** in units of **mm/microrad** for the orbit and **cm/microrad** for the dispersion component, as used inside the DFS program (DFS output)
- EIGENV-ORB : the orbit eigenvector matrix **U** (DFS output)
- EIGENV-COR : the corrector eigenvector matrix **V** (DFS output)

Items tagged with *DFS output* are not used as input for the response files, but as output from the dfs program (for further use...). Note that the response matrix must correspond to the **plane** indicated on the file header ! The two keyword lines are followed by **ncol** (binary) blocks of floats with length equal to **nrow**.

WARNING : In the present state, the STRING type data with the elements names is not (yet) used inside the program. The number and order of elements (monitors and correctors) must be exactly the same for the input data and for the matrices. A mismatch of the number of elements will results in an error, a mismatch of the order will yield absurd results. It is planned to improve this situation in the next version of the program.

References

- [1] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1987), 1st ed.
- [2] R. Assmann, P. Raimondi, G. Roy, J. Wenninger, *Emittance optimization with dispersion free steering at LEP*, Phys. Rev. ST Accel. Beams 3, 121001 (2000).
<http://prst-ab.aps.org/abstract/PRSTAB/v3/i12/e121001>
- [3] H. Grote and C. Iselin, CERN report CERN-SL/90-13 Rev. 3 (AP), 1990.