BNL-225644-2024-TECH

EIC-ADD-TN-094

SVD Orbit Correction in Zgoubi And Examples: SATURNE, RHIC, EIC RCS

F. Méot

May 2024

Electron-Ion Collider Project

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Nuclear Physics (NP)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No.DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

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, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use 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 or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

EIC TECHNICAL NOTE	NUMBER EIC-ADD-TN-094
AUTHOR **François Méot**	DATE May 2024

SVD Orbit Correction in Zgoubi And Examples: SATURNE, RHIC, EIC RCS

Abstract

This Tech. Note describes the SVD technique in the Lorentz force numerical solver zgoubi. It has been used as a tool to study the effect of vertical orbit perturbation on polarization transmission in RHIC, and for the study of the compensation of orbit defects resulting from stray fields in the EIC RCS [1].

[1] Priv. comm., Fanglei Lin, Vassiliy Morozov, SNS.

Contents

1 Introduction **3**

2 Method **3**

3 A Small 3 GeV Proton Ring **5**

4 RHIC **6**

5 EIC RCS **8**

1 Introduction

The SVD method is a convenient technique for orbit correction, of common use. It has been used in various designed studies using zgoubi, including beam lines, FFAG lines and rings, in EIC rings, etc. Pick-ups have provision to be multi-beam in this tool[2], to allow multiple beam correction (*e.g.* chromatic orbits, or momentum sets in fixed-field lattices).

The way the SVD command (SVDOC keyword in the following) operates is described here. Three examples are discussed: a small ring, SATURNE (105 m circumference), a larger ring, RHIC (3,830 m circumference), and a case of stray fields in drifts (the EIC RCS). All related input files, and many of the expected output files, can be found in sourceforge "exemple" repository, respectively at

https : //sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/KEYWORDS/SVDOC/SATURNE/

https : //sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/KEYWORDS/SVDOC/RHIC/

https : //sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/KEYWORDS/SVDOC/EIC_RCS/driftStrayFields/

This Tech. Note assumes some basic knowledge of the code. All details can be found in [1].

2 Method

SVDOC operates as a do-loop, looping on recording the response of pick-ups, to the excitation of correctors one by one. SVDOC has to be placed at the downstream end of the loop, *i.e.* at the bottom of the loop in question in the input data sequence. Once that do-loop is completed, the program execution pointer will quietly continue beyond SVDOC command in case anything follows.

SVDOC is expected to be operated for $N = 1$ or more closed orbits simultaneously. As zgoubi is a numerical solver (it integrates Lorentz force - and possibly Thomas-BMT - differential equation of motion), this requires defining N sets of initial coordinates differing by momentum offset $(x, x', y, y', \delta l, \delta p_{i=1,N}/p)$ (or, in zgoubi notations, $(Y, T, Z, P, S, D_{i=1,N})$), using OBJET[KOBJ=1, or KOBJ=2] for instance. This defines the N individual reference trajectories which the SVD process will handle.

$N > 1$ if different momenta are considered: chromatic orbits in a synchrotron ring, or closed orbits at different energies in an FFAG ring for instance. In that case SVDOC will assume that pick-ups are multiple-beam (each PU knows the actual position of each orbit). Correctors are global, each corrector applies one field value on all orbits at its location.

ERRORS or not ERRORS?

ERRORS may or may not be used. Two cases:

Case 1: the optical sequence already has a defect orbit included (quad positioning, field defects, or whatever else), then ERRORS is not necessary. Yet nothing precludes using ERRORS: this will generate errors on top of the existing defect orbit, once the A matrix is known, prior to launching the SVD correction process. Details below.

Case 2: the optical sequence does not include orbit defects. Then ERRORS can be used to inject errors. It will do so when A is known, prior to launching the SVD correction process. Details below.

Case 3: in case it is desired to assess the residual orbit for a set of ITER random error seeds, ERRORS will generate a new random defect to which the known matrix A is applied. Such repeat of the trials requires SVDOC[$JNF = 1,\ ITER \geq 1$].

FIT procedure:

In the case of a periodic structure, SVDOC can be preceded by *FIT*. *FIT* will find the closed orbit before launching the SVDOC procedure, and after completion of the latter. This final orbit is then expected to be noticeably damped, if SVDOC went well.

The way SVDOC operates is the following :

Step 1. Define/determine the reference trajectory (case of a beam line) or closed orbit (case of a periodic optical sequence) : the initial coordinate(s) in *OBJET* are considered the reference(s) - multiple references are allowed, see below. A reference does not

[2]This still requires some development in the pckup.f routine

have to be zero, however *FIT* can be used to find the reference orbit initial coordinates, in that case FIT should appear before SVDOC in the sequence. For instance,

(i) in a spectrometer or in a beam line, *FIT[IC=3]* would allow to determine the necessary initial coordinates (*FIT* variables) for a particle with a given momentum to acquire some requested radial excursion at some requested location down the line (*FIT* constraint(s));

(ii) if the sequence is periodic, *FIT[IC=3.1]* will find the closed orbit (possibly multiple closed orbits, *e.g.* in the case of a multiple-pass recirculating linac, or a fixed-field ring such as a cyclotron or FFAG).

Step 2. The algorithm computes the A matrix (in $A(\vec{x} - \vec{x}_0) = \vec{y} - \vec{y}_0$, with $\vec{x}$ the correctors signal, $\vec{y}$ the pickups signal, $\vec{x}_0$ and $\vec{y}_0$ initial states), by injecting a non-zero signal (angle) in each orbit corrector, one after the other successively, and saving the successive PU readings (as rows of the A matrix). There can be up to 20 different particles tracked (meaning 20 different reference orbits, and 20 times more columns (PUs) in A), each PU is "multiple-beam", capable of discriminating each orbit in the SVD process (case for instance of chromatic orbits in a synchrotron, or multiple momenta in a multiple-pass ERL, or multiple orbits in a scaling FFAG lattice).

Step 3. The inverse matrix A^{-1} is computed and logged to zgoubi.SVD_Amatrix.out (default name, can be changed), together with the A matrix and the result of an $A^{-1} \times A$ sanity test, expected to be the identity matrix.
At this stage, a version of the initial data file with correctors set, *i.e.* including existing initial orbit or ERRORS' defects and with correctors updated to the values resulting from SVDOC, is saved in zgoubi.SVD.out.dat (in a similar manner that the *FIT* procedures save a copy of the input data file, with variables updated, in zgoubi.FIT.out.dat).

Step 4. SVDOC applies the correction. A last pass through the sequence delivers the corrected orbit. The latter is logged in zgoubi.SVDOrbits.out.

Step 5. Optionally, SVDOC can carry on with (a user's specified number of) new error trials (for statistics on corrector settings and residual orbit, for instance), in the following way:
(i) ERRORS applies new set of orbit defects, this creates a new vector $\vec{y}$,
(ii) Step 4 above is repeated, it results in a new set of corrector values $\vec{x}$. These new corrector values and resulting PU records are stacked in zgoubi.SVDOrbits.out.

The process is the following for instance in the case of a periodic sequence:
- introduce *ERRORS[ONF=0]* in the optical sequence, it will be used by SVDOC to activate orbit defects. Note that *ERRORS* must be set to OFF (*ONF=0*), as (i) the first pass through the sequence is used to determine the $\vec{y}_0$ reference orbit mentioned earlier (using *FIT*, placed at the end of the sequence), (ii) the ON/OFF switch in *ERRORS* is managed by SVDOC in this procedure;
- *FIT* finds the closed orbit resulting from the defects generated by *ERRORS*; this orbit is recorded at the PUs that the SVD is told to use;
- SVDOC then sets the strengths in the correctors, using the SVD matrix;
- *FIT* finds the new orbit - presumably improved ! and delivers it (logged to zgoubi.SVDOrbits.out, together with the uncorrected initial orbit).

A recap of the output files produced by SVDOC:

file	content
zgoubi.SVD_Amatrix.out	contains A^{-1} and A matrices, as computed by SVDOC
zgoubi.SVD_PuCoList.out	Lists of pickups & their positions, and correctors & positions
zgoubi.SVDOrbits.out	In that order: pickup record (i) at first pass and (ii) right before svd, followed by corrector strengths, the latter followed by pickup records after correction
zgoubi.SVD.out.dat	a copy of zgoubi.dat sequence with defects set and updated corrector values as found by SVDOC, and with FIT (if used) and SVDOC commented. Running it is should show the corrected orbit

3 A Small 3 GeV Proton Ring

We use the polarized ion 3 GeV SATURNE lattice, here. The interest is mostly in the quick computation, given the small numbers of PUs and correctors, both 12 H and 12 V here (and easily changed to different numbers, if desired: the optical sequence has provision for up to 24 H and 24 V PUs and correctors).
A typical input data file in a first example (from the sub-folder [pathTo]/branches/exemples/SATURNE/random_quad_YZ-shift in sourceforge) is given in Tab. 1. The ring sequence is in SATURNE_RingSeq.inc, initially unperturbed in this example. The perturbation is caused by ERRORS, via x- and y-shift of lattice quadrupoles. Correctors are MULTIPOLs labeled HKIC and VKIC, see 'SVDOC'. Pick-ups are optical elements (or MARKERs) labeled PUH and PUV.
Outcomes of the SVD correction are displayed in the top right figure below. Obviously, HKIC and VKIC correctors are respectively located at x- and y-shifted quadrupoles (looking up SATURNE_RingSeq.inc confirms), this is the reason why the corrected orbit is zero: defects excited at QPF and QPD quadrupoles are locally compensated.
In a second example (from the sub-folder [pathTo]/branches/exemples/SATURNE/random_dipole_HV-Kicks/random_HVDEF_b1 in sourceforge), still using the unperturbed SATURNE_RingSeq.inc, a random dipole defect is injected using ERRORS, via H and V kicks (in MULTIPOLs labeled HDEF and VDEF). Outcomes are displayed in the bottom right figure below.

Table 1: A typical way to use the orbit correction algorithm. The native lattice orbit is zero (the lattice is not perturbed). After completion of the calculation of the A matrix, ERRORS generates a random orbit by shifting quadrupoles labeled QPF and QPD. That orbit is subsequently corrected.

```
SATURNE ring, circumf. = 105 m. Up to 12 PUH, 12 PUV, 12 H-correctors, 12 V-correctors.
'OBJET'
1000.0000
2
1 1
0.83051828 0.98542728  0.34682802  -2.6702268  0.0  1.0 'o'
1

!                       ERRORS is used, triggered by SVDOC after computation of A matrix,
'ERRORS'                                                 ! to generate a random defect.
0  4 234567     PRINT
MULTIPOL{QPD} 1 YS A U 0.  .25   1            ! quad dY causes dB_Z, horiz orbit defect,
MULTIPOL{QPD} 1 ZS A U 0.  .08   1           ! quad dZ causes dB_Y, vertic orbit defect.
MULTIPOL{QPF} 1 YS A U 0.  .1    1            ! quad dY causes dB_Z, horiz orbit defect,
MULTIPOL{QPF} 1 ZS A U 0.  .35   1           ! quad dZ causes dB_Y, vertic orbit defect.

'INCLUDE'
1
SATURNE_RingSeq.inc[SATURNE_Ring_S:SATURNE_Ring_E]

'FAISCEAU'  atFIT

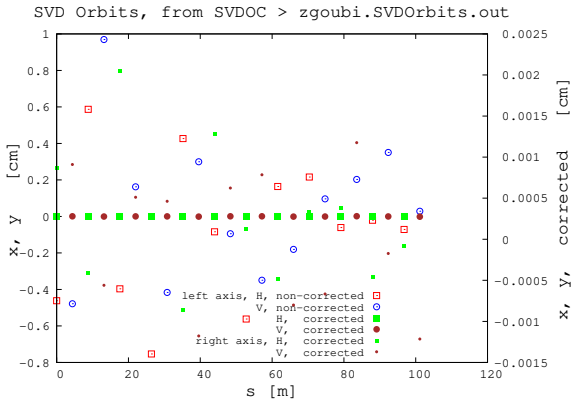
'FIT2'
4          ! final (NOT nofinal) is mandatory, it causes store of c.o. in zgoubi.SVD.out
1 30 0 [-13.9,13.9]   ! Vary Y0
1 31 0 [-19,19]   ! Vary T0
1 32 0 [-11.9,11.9]   ! Vary Z0
1 33 0 [-19,19]   ! Vary P0
4   1e-10
3.1 1 2 #End 0. 1. 0    ! Yfinal = Y0
3.1 1 3 #End 0. 1. 0    ! Tfinal = T0
3.1 1 4 #End 0. 1. 0    ! Zfinal = Z0
3.1 1 5 #End 0. 1. 0    ! Pfinal = P0

'DRIFT' ! just for greater accuracy on coordinates
0.000000
'FAISCEAU'  atSVDOC

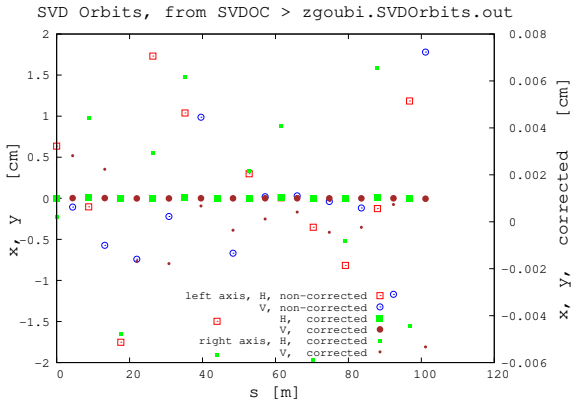
'SVDOC'
1  1
PUH{PUH} PUV{PUV}       ! Names of PU-H, -V, -HV families (max each =MPULAB/3=1 family),
CRH{HKIC} CRV{VKIC}                                    ! Name of corr-H, -V families.
1e-4     1e-4              ! Kick values (rad), used by SVDOC to generate the A matrix.
0 4 5  ! "1" triggers ERRORS; "4" data changed, following next 4 lines; 5 ERRORS trials.
ERRORS  12                                          ! parameter 12 in ERRORS: quad dY,
ERRORS  22                                          ! parameter 22 in ERRORS: quad dZ,
ERRORS  32                                          ! parameter 32 in ERRORS: quad dY,
ERRORS  42                                          ! parameter 42 in ERRORS: quad dZ.
!                                  step 1: at this stage, the refernece orbit is known,
!                             steps 2, 3: a loop on correctors determines PUs responses,
!    step 4: A^-1 is applied define corrector settings (come out null if reference orbit
! is null. When back down to FIT (from here), FIT will get the corrected periodic orbit.
!  step 5: process of applying A^-1 on random orbits repeats until ITER trials achieved.

'SYSTEM'
3
sed -n 's@KSVD@KSVD=1  #@g' gnuplot_Zfai_YZ-s.gnu
gnuplot <./gnuplot_SVDOrbits.gnu
gnuplot <./gnuplot_Zfai_YZ-s.gnu

'END'
```



Orbit records at 12 H + 12 V pickups, before and after correction, from zgoubi.SVD.out. ERRORS is used to excite 12 horizontal and 12 vertical random quadrupole shifts. Correctors are at the location of defects. The left vertical axis holds for both uncorrected and corrected records, the right vertical axis holds for the sole corrected records.



Orbit records at 12 H + 12 V pickups pickups, before and after correction. The initial optical sequence includes a native non-zero orbit in this case caused by local dipole kicks, 12 H and 12 V. HKIC and VKIC correctors are at the location of the latter.

4 RHIC

RHIC ring is considered here. The interest is mostly in the large number of local defects , pickups (180), and correctors (255). A typical input data file in a first example (from the sub-folder [pathTo]/branches/exemples/KEYWORDS/SVDOC/RHIC/ in sourceforge) is given in Tab. 2. The ring sequence is in rhic_sequence.inc, initially unperturbed in this example. The perturbation is caused by 'ERRORS', which excites random dipole field defects in lattice quadrupoles. Pickups are HMON and VMON families, correctors are MULTIPOLs labeled HKIC and VKIC, see 'SVDOC'. SVD correction outcomes are displayed below.

In a second example (from the sub-folder [pathTo]/exemples/KEYWORDS/SVDOC/RHIC/withOrbit/ in sourceforge), the native orbit is taken non-zero. This uses the perturbed ring sequence, rhic_SVDOC_withOrbit.dat input data file, which includes native random shifts of QF and QD arc quads. Outcomes of the SVD correction in that second example are displayed page 7.

Table 2: A typical way to use the orbit correction algorithm. The native lattice orbit is zero (the lattice is not perturbed). After completion of the calculation of the A matrix, ERRORS generates a random orbit by applying a defect dipole field in arc quadrupoles (labeled QF and QD). That orbit is subsequently corrected.

```
Injection, no IP bumps, D' lattice
 'OBJET'
79.366774592048841e3
2
1 1
0. 0. 0. 0.   0.00  1. 'o'
1

'ERRORS'
0  2 123456    PRINT                                        ! Set ERRORS to off for SVDOC.
MULTIPOL{QF} 1 BP A U 0. 2e-2 3              ! Test orbit in prior run: ought to be reasonable
MULTIPOL{QD} 1 BP A U 0. 2e-2 3                ! otherwise particles may be lost during SVDOC.

'INCLUDE'
1
./rhic_sequence.inc[RHIC$START:RHIC$END]

'FAISTORE'
1
zgoubi.fai none  ! all                                    Uncomment to get data stored.

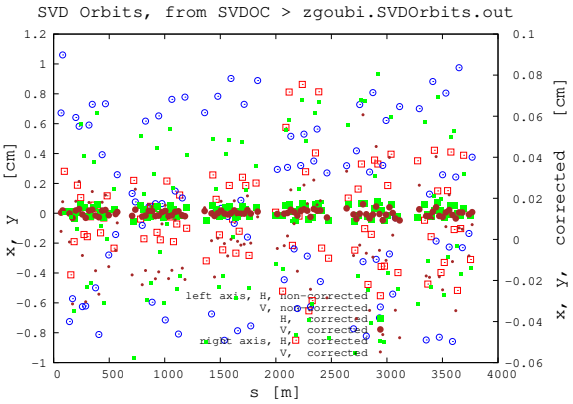
'FAISCEAU'
'FIT2'
4            ! final (NOT nofinal) is mandatory: causes store of actual c.o. in zgoubi.SVD.out.
1 30 0 [-5.9,5.9]   ! Vary Y0
1 31 0 [-9.9,9.9]   ! Vary T0
1 32 0 [-5.9,5.9]   ! Vary Z0
1 33 0 [-9.9,9.9]   ! Vary P0
4   1e-7
3.1 1 2 #End 0. 1. 0    ! Yfinal = Y0
3.1 1 3 #End 0. 1. 0    ! Tfinal = T0
3.1 1 4 #End 0. 1. 0    ! Zfinal = Z0
3.1 1 5 #End 0. 1. 0    ! Pfinal = P0

'SVDOC'
1  0
PUH{HMON} PUV{VMON}                  ! Names of PU-H, -V, -HV families (max each =MPULAB/3=1).
CRH{HKIC} CRV{VKIC}                                ! Name of corr-H, -V families; kick (rad).
1.e-3      1.e-3                            ! Kick values for computation of svd matrix (rad).
0  2  5  ! "1" triggers ERRORS; "2" data are changed prior to that, following 2 lines; 5 trials
ERRORS  12             ! parameter 12 in ERRORS: value of HKIC dipole field: will set H orbit,
ERRORS  22             ! parameter 22 in ERRORS: value of VKIC dipole field: will set V orbit.

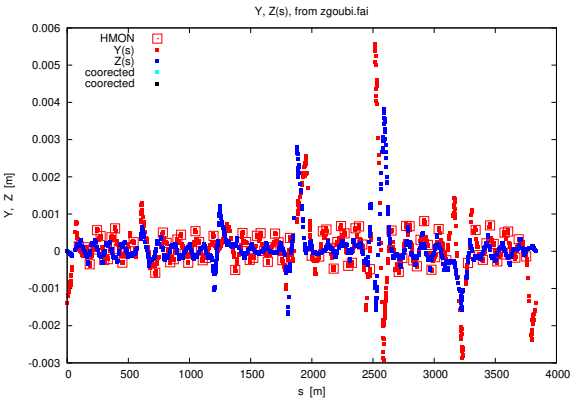
'SYSTEM'
1
gnuplot <./gnuplot_SVDOrbits.gnu
! okular gnuplot_SVDOrbits.eps  &

'END'
```

CASE OF ZERO NATIVE ORBIT:



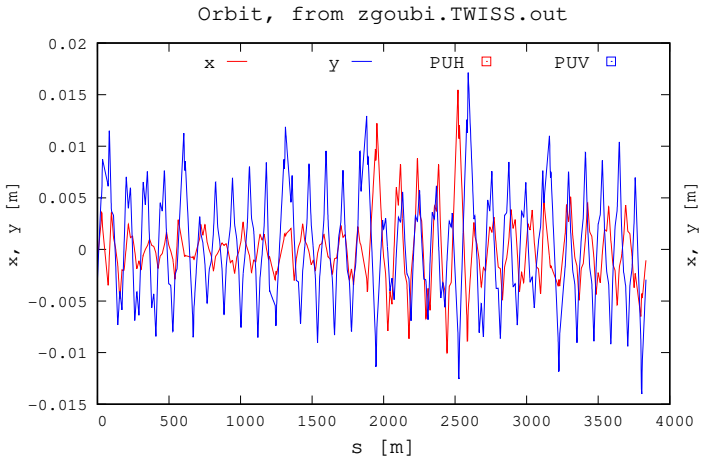
Orbit records at 180 pickups, before and after correction, from zgoubi.SVD.out. Orbit defect is caused by random dipole defects in QF and QD arc quads. Correctors are NOT at the location of defects. The left vertical axis holds for both uncorrected and corrected records, the right vertical axis holds for the sole corrected records.



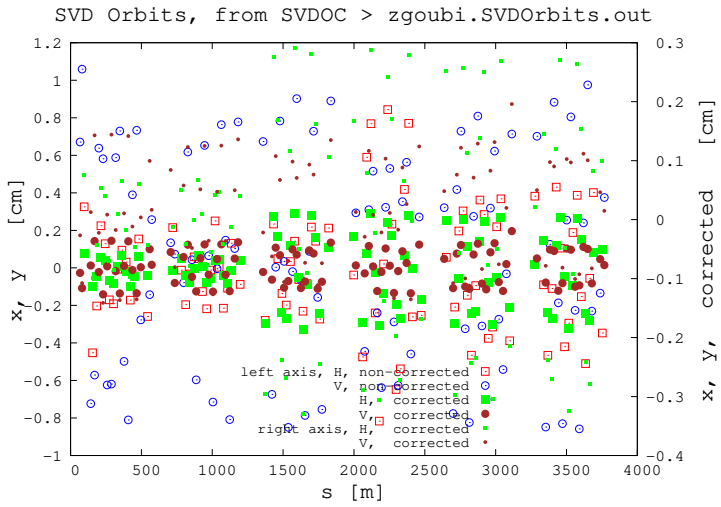
Orbit records after SVD correction, at all optical elements around RHIC, using FAISTORE[zgoubi.fai/all]. This includes the pickup data displayed in the figure above.

RHIC, cont'd

CASE OF NON-ZERO NATIVE ORBIT:



Initial non-zero orbit around RHIC, excited by native random shifts of QF and QD arc quads. This includes the pickup data (before correction) as displayed in the figure below.



Orbit records at 180 pickups, before and after correction, from zgoubi.SVD.out. The left vertical axis holds for both uncorrected and corrected records, the right vertical axis holds for the sole corrected records.

5 EIC RCS

The early version of the EIC RCS electron booster ring (400 MeV to 18 GeV) is considered here. The interest is mostly in the large number of local defects (stray fields in most of the drifts, by the 100s), pickups (696 H, 696 V), and correctors (696 H, 696 V). Note that, to save on CPU time (computation of the A matrix takes time with so many correctors) this example is run with half the instruments: 348 PUH, 348 PUV, 348 H correctors, 348 V correctors.
A typical input data file (from the sub-folder [pathTo]/branches/exemples/KEYWORDS/SVDOC/EIC_RCS/driftStrayFields/ in sourceforge) is given in Tab. 3. The ring sequence is in RCS_RingSequence.inc, initially perturbed as is, by stray fields in drifts. Correctors are MULTIPOLs labeled HKIC and VKIC, see 'SVDOC'.
Outcomes of the SVD correction are displayed below. Two different stray field amplitudes are SVDOC-ed, namely SCALING/DRIFT/0.03 and SCALING/DRIFT/0.08.

Table 3: A typical way to use the orbit correction algorithm. The native lattice orbit is NOT zero in this case, thus ERRORS is not needed (commented). After completion of the calculation of the A matrix, the existing native orbit is subsequently corrected.

```
! SVDOC-toring cloed orbit excited by stray fields.
'OBJET'
1334.2552920394014                                                  ! reference rigidity (kG.cm).
2
1   1
0. 0. 0. 0. 0.  1. 'o'
1
'PARTICUL'
0.51099892   1.60217653e-19   1.15965218076e-3   0.0   0.0
'SRLOSS'
0   !  .srloss
BEND MULTIPOL
1   123456
'SPNTRK'
3
'FAISCEAU'
'FAISTORE'
zgoubi.fai none
1

!'ERRORS'                              Commented, not needed as native orbit is non-null.
!0  1 123456                        It can be uncommented anyway, that will just change the
!MULTIPOL{DRIF} 1 BP A U 0. 2e-4 3          orbit which the A^-1 matrix will compensate.
!             It has to be added if SVDOC is requested to launch several independent trials.

'FAISCEAU'
'SCALING'
1   3
BEND
-1
1.3342552920E+00
1
MULTIPOL  QUAD  SEXT HKIC  VKIC
-1
1.3342552920E+00
1
MULTIPOL  DRIF
-1
0.03                                          ! Change this coefficient for orbit amplitude.
1

'INCLUDE'
1
RCS_RingSequence.inc[RCS_RING$START:RCS_RING$END]              ! EIC RCS optical sequence.

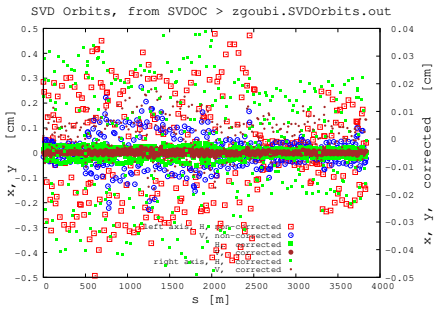
'FIT2'
4  noSYSout       ! final (NOT nofinal) is mandatory: causes store of c.o. in zgoubi.SVD.out.
1 30 0 [-5.9,5.9]                                                          ! Vary Y0,
1 31 0 [-9.9,9.9]                                                          ! Vary T0,
1 32 0 [-5.9,5.9]                                                          ! Vary Z0,
1 33 0 [-9.9,9.9]                                                          ! Vary P0.
4   1e-7
3.1 1 2 #End 0. 1. 0                                                    ! Yfinal = Y0,
3.1 1 3 #End 0. 1. 0                                                    ! Tfinal = T0,
3.1 1 4 #End 0. 1. 0                                                    ! Zfinal = Z0,
3.1 1 5 #End 0. 1. 0                                                    ! Pfinal = P0.

'OPTIONS'
1   1
WRITE ON

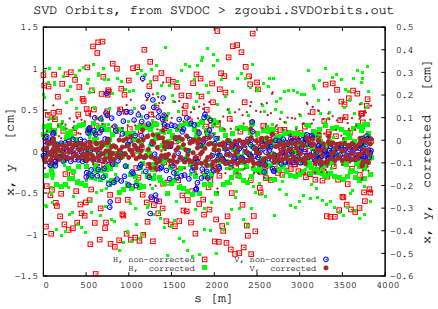
'SVDOC'
1  0  zgoubi.SVD.out
PUH{PUH} PUV{PUV}   ! Names of PU-H (696), -V (696), -HV (0) families (max each =MPULAB/3=1).
CRH{HKIC} CRV{VKIC}                    ! Name of corr-H (696), -V (696) families; kick (rad).
1.e-4    1.e-4                            ! Kick values for computation of svd matrix (rad).
0 2 5 ! '1' triggers ERRORS; '2' data are changed prior to that, following 2 lines; 5 trials.
ERRORS  12           ! parameter 12 in ERRORS: value of XHKIC dipole field: will set H orbit,
ERRORS  22            ! parameter 22 in ERRORS: value of VKIC dipole field: will set V orbit.
!                                                 step 1: A^-1 is applied to correctors.
!       Step 2: when back down to FIT (from here), FIT will get the corrected periodic orbit,

 'SYSTEM'
1
gnuplot <./gnuplot_SVDOrbits.gnu                  ! Will plot initial and SVDOC-ed orbits.

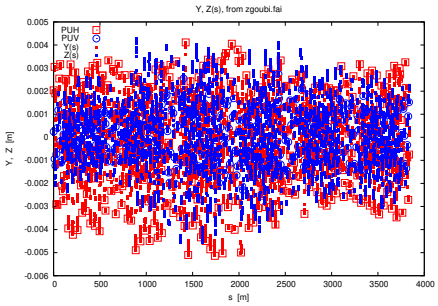
'END'
```



Orbit records at 2×348 pickups, before and after correction, from zgoubi.SVD.out. Orbit defect is caused by stray fields in drifts, DRIFT SCALING coefficient is 0.03 here. Correctors are NOT at the location of defects. The left vertical axis holds for both uncorrected and corrected records, the right vertical axis holds for the sole corrected records.



Same as above, yet with DRIFT SCALING coefficient set to 0.08, here.



Orbit records after SVD correction, at all optical elements around RCS, using FAISTORE[zgoubi.fai/all]. Case of DRIFT SCALING coefficient 0.08. This includes the pickup data displayed in the figure above.

References

[1] Zgoubi Users' Guide, F. Méot, Tech. Note CA-AP/ (2012), Rep. BNL-98726-2012-IR.
https://www.osti.gov/scitech/biblio/1062013-zgoubi-users-guide