DISCLAIMER

SRC Technical Notes are informal memos intended for internal communication and documentation of work in progress. These notes are not necessarily definitive and have not undergone a pre-publication review. If you rely on this note for purposes other than its intended use, you assume all risk associated with such use.
<table>
<thead>
<tr>
<th>University of Wisconsin-Synchrotron Radiation Center</th>
<th>File Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Note</td>
<td>SRC-112</td>
<td>1 of 1</td>
</tr>
</tbody>
</table>

**Subject:**
ADDENDUM TO SRC-95: "Program for Short-Time Particle Tracking of Injection"

**Name:** Bud Morin

**Date:** March 12, 1992

There has been a change in the original SRC-95 Technical Note. If you have obtained a prior to March 12, 1992, please request the new version from the SRC Secretarial Office.
Subroutine GETDX0

Subroutine to calculate DX0. Made by combining most of TRBACK with some code originally in LOSS. Only the 2x2 part of the transform matrix is used, corresponding to the situation with no kicker excitations. DX0 is later used by LOSS.

DX0 = a number proportional to the distance from the central orbit at the injection point to an aperture limit transformed back to the injection point.
XL = value of x at the aperture limit (mm). (Input variable)
XPL = a value of x' on the vertical line in phase space passing through x = XL; this point (XL, XPL) and the point (XL, -XPL) define the aperture line and they will be transformed back to the injection point to give two points defining a limit line in the phase space at the injection point. XPL in mrad. (Input variable).
TBK = the transfer matrix for transforming back to the injection point. Note: this is a 2x3 matrix and the i,3 component obtained from SYNC must be multiplied by 1000 to be used in this subroutine. (Input variable).
X1(I) = phase space vector after transforming (XL, XPL). (mm, mrad) (Output variable).
X2(I) = phase space vector after transforming (XL, -XPL). (mm, mrad) (Output variable).
U(I) = temporary phase space vector (x, x') (mm, mrad).
T(I,J) = temporary storage for X1 and X2; I = 1 for X1, I = 2 for X2.
I, J, K = indices

*******************************************************************************

subroutine GETDX0 (XL, XPL, TBK, DX0)

implicit none

integer I, J, K
real XL, XPL, TBK(2,3), X1(2), X2(2), U(2), T(2,2)
real DX0, F1, F2

U(1) = XL ! Init U
U(2) = XPL ! for X1
do I = 1, 2 ! Do the calculation twice; once for X1 and again for X2
do J = 1, 2 ! Calculate each component of Xi
T(I,J) = 0.0 ! Initialize
do K = 1, 2 ! Multiply 2x2 matrix on
T(I,J) = T(I,J) + TBK(J,K)*U(K) ! the vector
end do ! K
end do ! J
U(2) = -XPL ! Fix U for X2
end do ! I
! Repeat calculation for X2
X1(1) = T(1,1) ! Transfer
X1(2) = T(1,2) ! from T
X2(1) = T(2,1) ! to X1
X2(2) = T(2,2) ! and X2

F1 = X2(1) - X1(1) ! Get delta x
F2 = X2(2) - X1(2) ! and delta x'
DX0 = F1*X1(2) - F2*X1(1) ! Could simplify and not use
return ! F1 and F2, but want it to look
end ! like formula used in LOSS