DESCRIPTION
OF
LIELIB.FOR
É tienne Forest and Johan Bengtsson


June 1990



This library allows for the computation of normal form procedure on the ray representation of the map or on a single Lie exponent map (i.e. "pseudo-Hamiltonian). Of course, all these objects must be represented as Taylor series. The library links with the old DA-library of Berz.


The code can perform two types of normal form:

1) On symplectic maps using the Poisson bracket operator :f: defined through a function f as follows:

$$:f:g=[f,g]$$

The implementation covers the case of stable or unstable tunes as described in reference 1.


2) On non-symplectic maps using Lie operator $F_i\partial_i$ . Clearly, 1) is a special case of 2. This is useful in electron rings where the particle undergoes classical radiation. The map must have N eigenvalues of the form
$\exp(-a_i\pm 2\pi Q_i)$ where N is the dimension of the flow. The present document does not describe these routines.




Glossary


Ndim2    = maximum phase space dimension allowed = 6
Ndim     = Ndim2/2
Nd2      = phase space dimension used
Nd       = Nd2/2 number of degrees of freedom
No       = Degree of the differential algebra calculations
Nv ......    = Number of variables:  $Nv \geq Nd2$
Ndpt     = Position of the energy in a coasting beam analysis: Nd2 or Nd2-1. If      all the planes are oscillating then Ndpt=0


DA-map =: A collection of Nd2 polynomials

# Description of the Routines

## Initialization

SUBROUTINE LIEINIT(NO1,NV1,ND1,NDPT1,IREF1,NIS)

Initialize lielib and the DA package by calling DAINI.

NO1=No

NV1=Nv

ND1=Nd

NDPT1=Ndpt

IREF1=IREF files where resonances are written,

format:     number of resonances

$n_{x1}$           $n_{y1}$

$n_{x2}$           $n_{y2}$

...

$n_{xk}$ $n_{yk}$

0 if no resonance to be left in the map.

-1 no resonance and -p<tunes<p

NIS= Number of scratch IS(N) to be allocated

## Taylor Series Routine

SUBROUTINE ETALL(X,N)

Allocates X(N)

SUBROUTINE ETCCT(X,Y,Z)

Performs composition of two da-maps.

$Z = X \circ Y$

SUBROUTINE ETINI(X)

Creates the identity da-map.

X=E

SUBROUTINE ETINV(X,Y)

Creates the inverse da-map.

$Y = X^{-1}$

SUBROUTINE SIMIL(X,A,AI,Y)

Does a double composition of da-maps: hence a similarity transformation if the inverse of A is AI.

$Y = A \circ X \circ AI$

SUBROUTINE DALIND(H,RH,HT,RT,HR)

Does a linear combinations of da-maps.

HR=RH H + RT HT ;   RH and RT are real numbers.

SUBROUTINE DACOPD(H,HT)

HT=H ; H and HT are da-maps.

SUBROUTINE DACOND(H,R)

H(i)= DACON(H(i),R) ; i=1,nd2.
SUBROUTINE DACMUD(H,SCA,HT)
    DACMU(H(i),SCA,HT(I)) ; i=1,nd2.
SUBROUTINE TAKED(H,M,HT)

Projects the $M^{th}$ degree part H into HT. Both are da-maps.
SUBROUTINE TAKE(H,M,HT)

Projects the $M^{th}$ degree part H into HT. Both are single polynomials.
SUBROUTINE TRX(H,RH,Y)

RH=H $\circ$ Y;   RH is a polynomial and Y is a da-map.
SUBROUTINE DAREAD(H,ND1,MFILE,IPO)

Reads H(i), i=1,ND1 from mfile.

If IPO=0 it zeroes the zeroth order part of H(i).
SUBROUTINE DAPRID(H,N1,N2,MFILE)

Prints H(i), i=N1,N2 into mfile.


# Manipulation Routine of the Symplectic Case


SUBROUTINE LIEFACT(XY,X,H)

Produces the Dragt-Finn factorization of the da-map XY

XY= X exp(:$H_3$:) ...exp(:$H_{no}$:)

SUBROUTINE FEXPO(H,X,W,NRMIN,NRMAX,SCA,IFAC)

Produces a da-map W by acting on the da-map X with the following        definitions of
the Lie operators H. (H=$H_3$+ ...+$H_{no}$)

IFAC=1   Dragt-Finn map exp(:SCA $H_3$:) ...exp(:SCA $H_{no}$:)

IFAC=-1  Reversed Dragt-Finn map exp(:SCA $H_{no}$:) ...exp(:SCA$H_3$:)

IFAC=100                A single exponent map exp(:H:).


SUBROUTINE EXPND2(H,X,W,EPS,NRMAX)

Calls EXP1D until accuracy EPS is reached or until NRMAX Poisson
brackets.

Computes W= EXP(: H :) X as

$$W= \sum_{n=0}^{NRMAX} \frac{1}{n!} :H:^n \ X$$ ; W and X are da-maps.

SUBROUTINE EXP1D(H,X,Y,EPS,M,NON)

Computes Y= EXP(: H :) X ; X and Y are single polynomials.

IF M>2

It will expand the exponential as above to an order NRMAX given by

NRMAX=(NON-2)/(M-2)

NON = order of Taylor series + 1

M lowest degree of any monomial in H.

IF M $\leq$ 2

Computes enough Poisson brackets to reach an accuracy of EPS


SUBROUTINE LIE(H,X,Y,M,NON)

Computes Y= EXP(: H :) X ; X and Y are single polynomials by calling    EXP1D.
It will expand the exponential as above to an order NRMAX given by
NRMAX=(NON-2)/(M-2)
NON = order of Taylor series + 1
M lowest degree of any monomial in H.

# Manipulation Routine of the Symplectic Case of the Irwin Representation

SUBROUTINE IRWIN(H,X,NRMU,SEED,SCA,iq,iton1,iton2)
    Computes the Irwin factorization.
    H is the Dragt-Finn polynomial.
    X is the linear part of the map (a rotation).
    NRMU is the number of Irwin polynomial
    SCA is a scale (usually 1)
    iq is the file where X and the Irwin polynomials are printed in DA         format.
    iton1 and iton2 are the files for the angles and the Irwin      polynomials in a format used
by KICKMP.FOR, the tracking program.
    iton1 can also contain the linear transformation into Floquet         variables.

SUBROUTINE IRWINFAC(V,REL,NRMU,IFAC,NON,MFILE)
    Reads NRMU Irwin polynomials from MFILE.
    IFAC=1 ----> V= EXP(:KI$_1$:) ... EXP(:KI$_{NRMU}$:) REL

    IFAC=-1 ----> V= EXP(:-KI$_{NRMU}$:) ... EXP(:-KI$_1$:) REL

    NON is the highest degree of any Irwin monomial.
SUBROUTINE EXPO(KI,REL,V,IFAC,NON,NRMU)
    Same as above , in fact IRWINFAC calls EXPO

# Normal Form Routines in the Symplectic Case

SUBROUTINE GOFIX(XY,A1,A1I,NORD)
Finds the parameter dependent translation which brings XY to its fixed point
to order NORD.
    X = A1I ∘ XY ∘ A1 ; X(**0**)=**0** .
SUBROUTINE MAPNORM(X,FT,A2,A1,XY,H,NORD,KPMAX)
    Puts a map into normal form to order KPMAX.

    X = { exp(:FT$_{KPMAX}$:) ...exp(:FT$_3$:) A2 A1 }$^{-1}$ exp(:H$_2$:) ...exp(:H$_{NO}$:)
        exp(:FT$_{KPMAX}$:) ...exp(:FT$_3$:) A2 A1 +Order(KPMAX+1)

    XY = exp(:FT$_{KPMAX}$:) ...exp(:FT$_3$:) A2 A1 X

                { exp(:FT$_{KPMAX}$:) ...exp(:FT$_3$:) A2 A1 }$^{-1}$+ ...Order(No+1)
    Nord : see subroutine GOFIX.
SUBROUTINE HAMNORM(X,FT,A2,A1,H,NORD,KPMAX)
    Normalizes the map exp(:X:) using the pseudo-hamiltonian X.

SUBROUTINE DHDJ(H,T)

Computes the tunes of the map exp(:H(**J**):).

T(i) proportional to $\partial H/\partial J_i$    i=1,nd in the cartesian basis.

T(i+nd) proportional to $\partial H/\partial J_i$    i=nd+1,nd2 in the resonance basis

If Ndpt not zero then $J_{nd}$ = energy and the nd entry above must be the    path in

Floquet space (i.e. momemtum compaction and its nonlinear        parts)

# Manipulation Routines of the Symplectic Normal Form

If no analysis of the map has been performed, stability in all planes is assumed.

SUBROUTINE CTOR(C1,R2,I2)

The cartesian to resonance change of basis

C1 = R2 + i I2 ;    $i=\sqrt{-1}$

R2---> cosine terms

I2---> sine terms

SUBROUTINE RTOC(R1,I1,C2)

The resonance to cartesian change of basis

R1 + i I1 = C2    $i=\sqrt{-1}$

R1---> cosine terms

I1---> sine terms

SUBROUTINE H2PLUG(H,ANG,IPO)

Changes or reads the $H_2$ part of H(**J**)

IPO=-1        Reads ang(i) ; i=1,nd.
IPO=0         Plugs ang(i) into $H_2$

IPO=1         Adds ang(i) to  $H_2$.

st(i) = 1 if $i^{th}$ plane is stable, 0 otherwise.

# Non Symplecitic Case

SUBROUTINE DIFD(H,V)

Takes a symplectic Lie operator :H: and creates the equivalent flow        operator
:H: = $V(i)\partial_i$ i=1,nd2

SUBROUTINE DAFLO(H,X,Y,N)
_____Takes a flow operator $H(i)\partial_i$ i=1,N and acts on the polynomial X:

Y=$H(i)\partial_i$ Y

SUBROUTINE FLOFACT(XY,X,H)
SUBROUTINE FEXFLO(H,X,W,NRMIN,NRMAX,SCA,IFAC)
SUBROUTINE EXPF1D(H,X,Y,EPS,M,NON)
SUBROUTINE EXPFND2(H,X,W,EPS,NRMAX)
SUBROUTINE LIEFLO(H,X,Y,M,NON)

# Normal Form Routines in the Non Symplectic Case

SUBROUTINE MAPNORMF(X,FT,A2,A1,XY,H,NORD,KPMAX,ISI)

# Manipulation Routines of the Non Symplectic Normal Form

SUBROUTINE CTORFLO(C1,R2,I2)
SUBROUTINE RTOCFLO(R1,I1,C2)
SUBROUTINE H2PLUFLO(H,ANG,RA,IPO)

SUBROUTINE INITPERT(ST,IH,ANG,RA)

    Creates the arrays of eq. 14c

    $ST(i) = 1 \Rightarrow$ stable $i^{th}$ plane
    Ang(k) contains the $m_k$ of eq. 15a.

    RA(k) contains the damping term for the nonsymplectic case
    IH=0 => map normalization and IH=1 pseudo-Hamiltonian       normalization.
    Reads from file 3 the resonances to be left untouched.

SUBROUTINE CPART(H,CH)

    CH= $f_p$ H as in eq. 22. uses function REXT.for p.

DOUBLE PRECISION FUNCTION REXT(J)

    See above subroutine

SUBROUTINE ROT(RO,ANG)

    RO= $\exp(:f_2:)$ where $f_2$ is given by 15c.

    Ang(k) contains the $m_k$ of eq. 15a.

SUBROUTINE ROTI(ROI,ANG)

    The inverse of the above routine: ROI= $\exp(:-f_2:)$

SUBROUTINE HYPER(A,CH,SH)

    Computes Cosh(A) and Sinh(A).Used in ROT and ROTI.

SUBROUTINE ETCTR(X)

    X= $\neg_r$ of eq. 27a.

SUBROUTINE ETRTC(X)

    X= $\neg_r^{-1}$ of eq. 27c

SUBROUTINE ETCJG(X)

    X= ¡ of eq. 27b

SUBROUTINE CTOI(F1,F2)

    Going from the cartesian into the Irwin representation.
              F1--> F2= $f_p$ F1 $\circ$ $\neg_r$

SUBROUTINE ITOC(F1,F2)

    The inverse of the above:

    F1-->F2=$f_p$ (F1 $\circ$ $\neg_r^{-1}$)

DOUBLE PRECISION FUNCTION GAM(J)

    a(j) of eq.24a (or appendix B-5)

DOUBLE PRECISION FUNCTION GBM(J)

    b(j) of eq.24b (or appendix B-5)

DOUBLE PRECISION FUNCTION TUER(J)

    Creates the Kernel of $\mathbf{T}^{-1}$; see below.

SUBROUTINE INTD(V,H)

    Computes the Lie polynomial H such that:
    [H,$\mathbf{x}$]= V ; See eqs. 34 through 37.

DOUBLE PRECISION FUNCTION DLIE(J)

    Computes s($\mathbf{j}$) of eq. 36.

SUBROUTINE TINV(H,FT)

Input    H= $H_{Range} + H_{Kernel}$

Output    FT = $\mathbf{T}^{-1}$ $H_{Range}$ and H = $H_{Kernel}$ ; the Kernel is selected in subroutine   INITPERT.
The function TUER is used with DACFU of Berz' Package to create $H_{Kernel}$ .

This calculation is done in the Irwin representation.

SUBROUTINE ORDERMAP(H,FT,X,ANG,KPMAX)

Recursive loop of section 6.
X= R $N_0$ of eq. (39)

$N_W$ of eq. 40c is given by H and A of 40b by FT.

SUBROUTINE ORDERHAM(H,T,FT,KPMAX)

Recursive loop of appendix B.
Use in HAMNORM.

SUBROUTINE MIDBLOCK(C,A2,A2I,Q,ST)

Translate the linear da-maps C,A2 and A2i into regular FORTRAN          arrays. Then it calls the next routine MAPNORML

A1I ∘ C ∘ A1 = a rotation of angles q(i) i=1,nd

ST(i)=0,1 => unstable , stable.

SUBROUTINE MAPNORML(SA,SAI,CR,CM,P,ST)

The matrix relation  CR= SAI CM SA is established, where CR is a          rotation of angles P(i). (ST(i) as above)

SUBROUTINE MULND2(rt,r)

Multiplies matrices in MAPNORML

SUBROUTINE ANALIE(H,FT)

Called in ORDERMAP or ORDERHAM to transform the Lie polynomial H
and FT into the Irwin basis. It then calls TINV and reconverts the          results into the Cartesian basis.

Same input-output structure as in TINV.

Modified Eispac routines by Filippo Neri also used in LIELIB.FOR

    subroutine eig6(fm,reval,aieval,revec,aievec)
       This routine computes the eigenvalues of the matrix fm and the       eigenvectors.
It calls the following routines:

    subroutine ety(nm,n,low,igh,a,ort)
    subroutine etyt(nm,n,low,igh,a,ort,z)
    subroutine ety2(nm,n,low,igh,h,wr,wi,z,ierr)
    subroutine etdiv(a,b,c,d,e,f)
    subroutine sympl3(m)

Acknowledgements

The Differential Algebra package we are using is the old version of Martin Berz to whom we are very grateful. We are also grateful to Filippo Neri for his work on the old Eispac routines.

Reference

1. E. Forest, M Berz and J. Irwin, Normal Form Methods for Complicated Periodic Systems: A Complicated Solution Using Differential Algebra and Lie Operators, Part. Accel. , 1989 , Vol. 24, p. 91-107.