Perturbation theory adapted to tracking codes

TPSA based perturbation theory and analytical results

Étienne Forest[1,2]

[1] High Energy Research Accelerator Organization Tsukuba, Japan

[2] Department of Accelerator Science
Graduate University for Advanced Studies, Hayama, Japan

December 2017

Outline

What is Perturbation Theory and why do we care?

- Perturbation theory is an attempt to transform (approximately) a difficult problem into a simpler one usually solvable
- The simpler problem can be:
  1. an extension of a linear problem
  2. a pendulum-like problem (one resonance map)
- It is perhaps possible to find formulae of quantities defined in perturbation theory and correlate their predictions to observable tracking results.

P.S. The real problem, that is to say the map produced by tracking, can be more complex. It can have chaos, multiple resonances, etc. . .

Wonderful Magnets Inc.

A company with secret magnet designs

Let us imagine the following:

- W.M.Inc. has its own proprietary version of all the standard magnets: quadrupoles, bends, etc ...
- W.M.Inc. keeps all internal aspects of the magnets as secret as it can
- W.M.Inc. provides a tracking/design code where all its proprietary magnets are implemented as well as standard magnets
- The code is equipped with TPSA (Truncated Power Series Algebra) and therefore is capable of tracking Taylor series representation of all the quantities the usual “double precision” code tracks
- You decide to design and/or test a lattice using the miraculous magnets of W.M.Inc.
- You must use their code for the work since no one knows what is inside their magnets

Question: Can you compute lattice functions, synchrotron integrals, tunes shifts, phase advances, etc...despite having no knowledge of the magnets inner structure?

Example of things we may want to compute z_my_nonlinear_twiss_phase_average_x_wm.f90

1. Change of tune with respect to some quadrupole family: $\frac{d\mu}{dk_Q}$
2. The average of x (say time average) as a function of the invariant $\frac{d\langle x\rangle}{dJ}$
3. Total linear phase advance

All these can be evaluated with TPSA. So if TPSA is implemented in the secret code of the company, we can get the results without knowing the exact nature of W.M.Inc.'s wonderful magnets!

Let us try on the following lattice:

```
!   mad-x lattice
!   L : drift, L= 1.0;
!   alpha= 0.314159265358979;
! QF : SBEND,L= 1.0, ANGLE=ALPHA,k1=1.0;
! QD : SBEND,L= 1.0, ANGLE=ALPHA,k1=-1.0;
!   sf  : sextupole,  K2= 2.0;
!lattice : LINE=  10*(QF,sf,L,QD,sf,L);
```

Model: 1-d-f Ideal Magnets in the small angle approximation!

$$H = \frac{p_x^2}{2\left(1+\delta\right)} \qquad + \left(b_0 - h\right) x + \frac{hb_0}{2} x^2 - \frac{hx\delta}{2} + k_Q\left(s\right) \frac{x^2}{2} + k_S\left(s\right) \frac{x^3}{6} + \cdots$$

$$\frac{dx}{ds} = [x, H] = \frac{p_x}{\left(1+\delta\right)}$$

$$\frac{dp_x}{ds} = [p_x, H] = -\left(b_0 - h\right) - \left(hb_0 + k_Q\left(s\right)\right) x \; - \frac{1}{2} k_S\left(s\right) \; x^2 + \cdots$$

Implementation

```
dl=mag%L/mag%n

   z=r%z

z(1)=z(1)+DL/2.d0*z(2)/(1.d0+z(3))*(1.0_dp+e1*mag%bnr(1))   <-- drift

fac=1.0_dp
do j=0,nmul
 z(2)=z(2)-mag%bn(j)*dl*z(1)**(j)/fac    <-- kick
  fac=fac*(j+1)
enddo

z(1)=z(1)+DL/2.d0*z(2)/(1.d0+z(3))*(1.0_dp+e1*mag%bnr(1))   <-- drift
```

One step of a 2^{nd} order integrator

Model: 1-d-f Ideal Magnets in the small angle approximation!

$$H = \frac{p_x^2}{2(1+\delta)} (1+\epsilon k_Q(s)) + (b_0 - h)\, x + \frac{hb_0}{2} x^2 - \frac{hx\delta}{2} + k_Q(s)\frac{x^2}{2} + k_S(s)\frac{x^3}{6} + \cdots$$

$$\frac{dx}{ds} = [x, H] = \frac{p_x}{(1+\delta)} (1+\epsilon k_Q(s)) \longleftarrow \text{This is the secret term of mythical W.M.Inc. company!}$$

$$\frac{dp_x}{ds} = [p_x, H] = -(b_0 - h) - (hb_0 + k_Q(s))\, x - \frac{1}{2} k_S(s)\; x^2 + \cdots$$

Implementation

```
dl=mag%L/mag%n

   z=r%z

z(1)=z(1)+DL/2.d0*z(2)/(1.d0+z(3))*(1.0_dp+e1*mag%bnr(1))   <--- drift

fac=1.0_dp
do j=0,nmul
 z(2)=z(2)-mag%bn(j)*dl*z(1)**(j)/fac     <--- kick
  fac=fac*(j+1)
enddo

z(1)=z(1)+DL/2.d0*z(2)/(1.d0+z(3))*(1.0_dp+e1*mag%bnr(1))   <--- drift
```

One step of a 2^{nd} order integrator

Results from the “experts” on standard magnets

1. Change of tune with respect to some quadrupole family:
   - $$d\nu = \frac{1}{4\pi}\oint_0^C \beta k_Q \, ds$$
2. The average of x as a function of the invariant
   - $$\frac{\partial \langle \widetilde{x} \rangle_s}{\partial J} = \frac{\beta_s^{1/2}}{2\left(1-\cos\left(\mu\right)\right)} \oint_0^C \left(-\sin\left(\mu_{s\sigma}\right) + \sin\left(\mu_{s\sigma} - \mu\right)\right) \beta_\sigma^{3/2} k_S(\sigma) d\sigma$$
3. Total linear phase advance
   - $$\nu = \frac{\mu}{2\pi} = \frac{1}{2\pi}\oint_0^C \frac{ds}{\beta}$$

Results from the code and the standard formulae!

1. Change of tune with respect to some quadrupole family: we select the first QD
   - Code (TPSA): $d\nu = 0.0936\ dk$
   - Formula evaluated by code: $d\nu = 0.0597\ dk$

$$d\nu = \frac{1}{4\pi}\oint_0^C \beta k_Q\, ds$$

2. The average of x as a function of the invariant
   - Code (TPSA): $\langle x\rangle_0 = -9.78\ J$
   - Formula evaluated by code: $\langle x\rangle_0 = -9.78\ J$

$$\frac{\partial\langle\tilde{x}\rangle_s}{\partial J} = \frac{\beta_s^{1/2}}{2\left(1-\cos\left(\mu\right)\right)}\oint_0^C \left(-\sin\left(\mu_{s\sigma}\right)+\sin\left(\mu_{s\sigma}-\mu\right)\right)\beta_\sigma^{3/2} k_S(\sigma) d\sigma$$

3. Total linear phase advance
   - Code (TPSA): $\mu = 3.44$
   - Formula evaluated by code: $\mu = 3.92$
   -

$$\nu = \frac{\mu}{2\pi} = \frac{1}{2\pi}\oint_0^C \frac{ds}{\beta}$$

The experts at W.M.Inc. give us new formulae!

1. Change of tune with respect to some quadrupole family: we select the first QD
   - Code (TPSA): $d\nu = 0.0936\ dk$
   - Formula evaluated by code: $d\nu = 0.0936\ dk$

$$d\nu = \frac{1}{4\pi} \oint_0^C (\beta + \overset{?}{\epsilon\gamma})\, k_Q ds$$

2. The average of x as a function of the invariant
   - Code (TPSA): $\langle x \rangle_0 = -9.78\ J$
   - Formula evaluated by code: $\langle x \rangle_0 = -9.78\ J$

$$\frac{\partial \langle \widetilde{x} \rangle_s}{\partial J} = \frac{\beta_s^{1/2}}{2\left(1 - \cos\left(\mu\right)\right)} \oint_0^C \left(-\sin\left(\mu_{s\sigma}\right) + \sin\left(\mu_{s\sigma} - \mu\right)\right) \beta_\sigma^{3/2} k_S(\sigma) d\sigma$$

3. Total linear phase advance
   - Code (TPSA): $\mu = 3.44$
   - Formula evaluated by code: $\mu = 3.44$
   -

$$\nu = \frac{\mu}{2\pi} = \frac{1}{2\pi} \oint_0^C \frac{ds}{\beta} (1 + \overset{?}{\epsilon k_Q})$$

Questions for the lecture

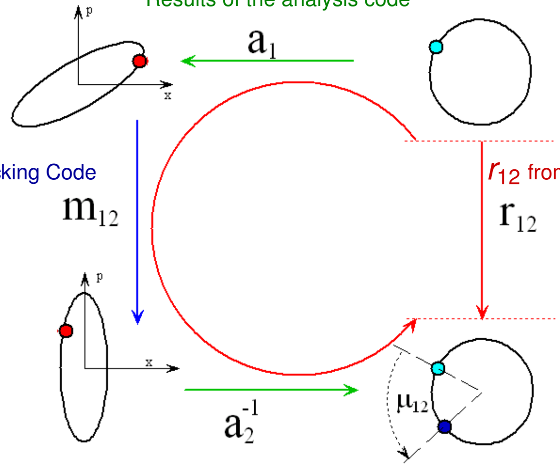
- Why are the calculations using the code (TPSA) correct?
  - This is a consequence of the lecture on TPSA
  - If not obvious, you have to get a sense of why it must be obvious
  - No knowledge of the details of any calculation necessary
  - Analytical W.M. formulae are only correct under some conditions
  - They are “difficult” to derive
- Why are some formulae still good and some are bad?
  - We will look at pure TPSA calculations
  - We will look at some formulae and how to sometimes connect them to pure TPSA calculations

The Mother of all Diagrams: Phase advance loop

Results of the analysis code

m_{12} from the Tracking Code

r_{12} from the Tracking Code



Example Programme

z_my_nonlinear_twiss_phase_average_x.f90

The Mother of all Diagrams

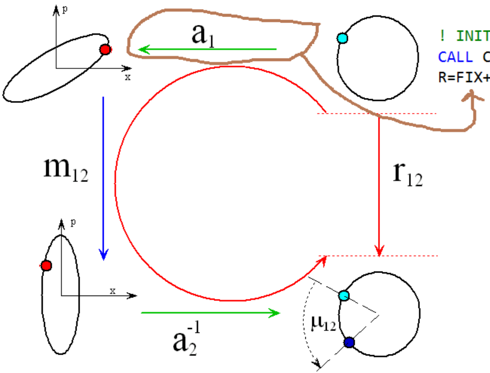
Finding the one-turn map

```
DELTA_IS_3RD_PARAMETER=.TRUE.          ←— X(3) will be the third variable of TPSA

ORDER_OF_TAYLOR=3           ←— The order of TPSA will 3

FIX=0.0_DP  ! FIXED POINT
CALL FIND_CLOSED_ORBIT( FIX,LATTICE,1)  ←— Finding the closed orbit: not necessarily (0,0)
WRITE(6,*) FIX(1:2)
! INITIALIZE THE RAY AS  -->
!RAY = FIXED POINT + IDENTITY (TAYLOR MAP)
ID=1
R=FIX+ID        ←— Ray is equal to the closed orbit + identity

!  COMPUTING A ONE-TURN MAP TO ORDER MY_ORDER

CALL TRACK_LATTICE( R,LATTICE,1,1)  ←— Computing the ray for one turn
M=R  ←— Putting the ray into a map: one-turn map created

! NORMALIZING THE MAP
CALL NORMALISE(M,NORMAL)       ←— Normalising the map : M = A_cs R A_cs^-1
A_CS=NORMAL%A_T
```

Normalising the map: $M = A_{cs} R A_{cs}^{-1}$

The Mother of all Diagrams

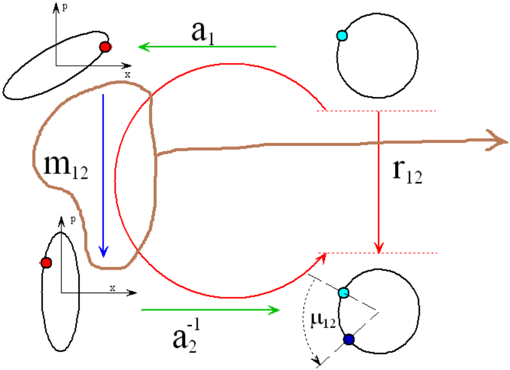
Initialising the Twiss loop!




```
! INITIALIZE THE RAY AS --> RAY = FIXED POINT + A
CALL CANONISE( NORMAL%A_T ,A_CS )
R=FIX+A CS
```

The Mother of all Diagrams

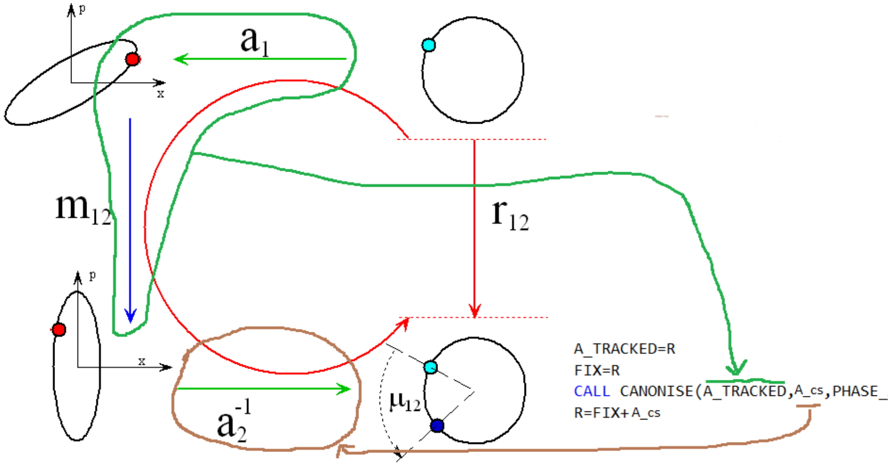
Tracking through a magnet




```
DO J=1,SIZE(LATTICE)
CALL  TRACK_MAGNET( R,LATTICE(J) )
        .
        .
        .
ENDDO
```

The Mother of all Diagrams

Extracting the phase advance



The Mother of all Diagrams

Doing a calculation in the Twiss loop

```
DO J=1,SIZE(LATTICE)

CALL TRACK_MAGNET( R,LATTICE(J) )

 A_TRACKED=R
 FIX=R
 CALL CANONISE(A_TRACKED,A_2,PHASE_ADVANCE0=PHASE_ADVANCE)
 R=FIX+A_2

BETAX=(A_CS%V(1).INDEX.1)**2+(A_CS%V(1).INDEX.2)**2
ALPHAX=-((A_CS%V(1).INDEX.1)*(A_CS%V(2).INDEX.1)+(A_CS%V(1).INDEX.2)*(A_CS%V(2).INDEX.2))

DX_AVERAGE_DJ=(BETAX)**1.5_DP*LATTICE(J)%BN(2)/4.0_DP  &
*(-SIN(PHASE_ADVANCE*TWOPI)+SIN((PHASE_ADVANCE-NORMAL%TUNE)*TWOPI))  &
/(1.0_DP-COS(NORMAL%TUNE*TWOPI))  + DX_AVERAGE_DJ

ENDDO

DX_AVERAGE_DJ=DX_AVERAGE_DJ*SQRT(BETAX)
```

$\longleftarrow \beta_x = A_{11}^2 + A_{12}^2$

$\alpha_x = -A_{11}A_{21} - A_{12}A_{22}$

$$\frac{\partial\langle\widetilde{x}\rangle_s}{\partial J} = \frac{\beta_s^{1/2}}{2\left(1-\cos\left(\mu\right)\right)} \oint_0^C \left(-\sin\left(\mu_{s\sigma}\right) + \sin\left(\mu_{s\sigma}-\mu\right)\right) \beta_\sigma^{3/2} k_S(\sigma) d\sigma$$

The Numerical Results
The one-turn map

```
The one-turn map :  x  and  p

  (1,0,0)-0.2020E+01    (1,0,0)-0.7305E+00
  (0,1,0) 0.1811E+01    (0,1,0) 0.1597E+00
  (0,0,1) 0.1975E+01    (0,0,1) 0.9450E+00
  (2,0,0) 0.7423E+02    (2,0,0) 0.3697E+02
  (1,1,0)-0.2504E+03    (1,1,0)-0.1451E+03
  (0,2,0) 0.1783E+03    (0,2,0) 0.1196E+03
  (1,0,1) 0.3130E+01    (1,0,1) 0.1923E+02
  (0,1,1) 0.7377E+01    (0,1,1)-0.1432E+02
  (0,0,2)-0.8484E+01    (0,0,2)-0.1054E+02
  (3,0,0)-0.4292E+04    (3,0,0)-0.2717E+04
  (2,1,0) 0.1924E+05    (2,1,0) 0.1175E+05
  (1,2,0)-0.2997E+05    (1,2,0)-0.1758E+05
  (0,3,0) 0.1602E+05    (0,3,0) 0.8910E+04
  (2,0,1) 0.3406E+04    (2,0,1) 0.2582E+04
  (1,1,1)-0.7892E+04    (1,1,1)-0.6107E+04
  (0,2,1) 0.4963E+04    (0,2,1) 0.3860E+04
  (1,0,2)-0.1317E+04    (1,0,2)-0.1146E+04
  (0,1,2) 0.1269E+04    (0,1,2) 0.1290E+04
  (0,0,3) 0.1711E+03    (0,0,3) 0.1588E+03
```

$$M(0,0,\delta) = \left(1.975\,\delta \;-\; 8.484\,\delta^2 + \cdots, 0.945\,\delta \;-\; 10.45\,\delta^2 + \cdots, \delta\right)$$

The Numerical Results

The one-turn map expressed around the δ-dependent closed orbit

$$\bar{z}_1 = z_1 + \eta_1(z_3)$$
$$\bar{z}_2 = z_2 + \eta_2(z_3)$$

```
M=NORMAL%disp**(-1)*M*NORMAL%disp   and   NORMAL%disp
 (1,0,0)-0.2020E+01   (1,0,0)-0.7305E+00   (1,0,0) 0.1000E+01   (1,0,0) 0.0000E+00
 (0,1,0) 0.1811E+01   (0,1,0) 0.1597E+00   (0,1,0) 0.0000E+00   (0,1,0) 0.1000E+01
 (0,0,1) 0.0000E+00   (0,0,1) 0.0000E+00   (0,0,1) 0.8732E+00   (0,0,1) 0.3655E+00
 (2,0,0) 0.7423E+02   (2,0,0) 0.3697E+02   (2,0,0) 0.0000E+00   (2,0,0) 0.0000E+00
 (1,1,0)-0.2504E+03   (1,1,0)-0.1451E+03   (1,1,0) 0.0000E+00   (1,1,0) 0.0000E+00
 (0,2,0) 0.1783E+03   (0,2,0) 0.1196E+03   (0,2,0) 0.0000E+00   (0,2,0) 0.0000E+00
 (1,0,1) 0.4123E+02   (1,0,1) 0.3076E+02   (1,0,1) 0.0000E+00   (1,0,1) 0.0000E+00
 (0,1,1)-0.8096E+02   (0,1,1)-0.5359E+02   (0,1,1) 0.0000E+00   (0,1,1) 0.0000E+00
 (0,0,2) 0.0000E+00   (0,0,2) 0.0000E+00   (0,0,2)-0.1087E+01   (0,0,2)-0.3975E+00
 (3,0,0)-0.4292E+04   (3,0,0)-0.2717E+04   (3,0,0) 0.0000E+00   (3,0,0) 0.0000E+00
 (2,1,0) 0.1924E+05   (2,1,0) 0.1175E+05   (2,1,0) 0.0000E+00   (2,1,0) 0.0000E+00
 (1,2,0)-0.2997E+05   (1,2,0)-0.1758E+05   (1,2,0) 0.0000E+00   (1,2,0) 0.0000E+00
 (0,3,0) 0.1602E+05   (0,3,0) 0.8910E+04   (0,3,0) 0.0000E+00   (0,3,0) 0.0000E+00
 (2,0,1)-0.8055E+03   (2,0,1)-0.2417E+03   (2,0,1) 0.0000E+00   (2,0,1) 0.0000E+00
 (1,1,1) 0.3807E+04   (1,1,1) 0.1559E+04   (1,1,1) 0.0000E+00   (1,1,1) 0.0000E+00
 (0,2,1)-0.3638E+04   (0,2,1)-0.1723E+04   (0,2,1) 0.0000E+00   (0,2,1) 0.0000E+00
 (1,0,2) 0.1451E+03   (1,0,2) 0.4358E+02   (1,0,2) 0.0000E+00   (1,0,2) 0.0000E+00
 (0,1,2) 0.1001E+03   (0,1,2) 0.1475E+03   (0,1,2) 0.0000E+00   (0,1,2) 0.0000E+00
 (0,0,3) 0.0000E+00   (0,0,3)-0.1137E-12   (0,0,3) 0.1848E+01   (0,0,3) 0.5969E+00
```

$M(0,0,\delta) = (0,0,\delta)$

The Numerical Results

The one-turn map around the δ-dependent orbit and linearly normalised

$$A_L : \begin{array}{l} \bar{z}_1 = \sqrt{\beta_x(z_3)}\, z_1 \\ \bar{z}_2 = -\alpha_x(z_3)/\sqrt{\beta_x(z_3)}\, z_1 + z_2/\sqrt{\beta_x(z_3)} \end{array}$$

```
M=NORMAL%A_L**(-1)*M*NORMAL%A_L     and    NORMAL%A_L
 (1,0,0)-0.9301E+00   (1,0,0)-0.3674E+00   (1,0,0) 0.2220E+01   (1,0,0) 0.1336E+01
 (0,1,0) 0.3674E+00   (0,1,0)-0.9301E+00   (0,1,0) 0.0000E+00   (0,1,0) 0.4504E+00
 (2,0,0)-0.2644E+02   (2,0,0) 0.1459E+01   (2,0,0) 0.0000E+00   (2,0,0) 0.0000E+00
 (1,1,0)-0.1611E+02   (1,1,0) 0.4536E+02   (1,1,0) 0.0000E+00   (1,1,0) 0.0000E+00
 (0,2,0) 0.1630E+02   (0,2,0) 0.5536E+01   (0,2,0) 0.0000E+00   (0,2,0) 0.0000E+00
 (1,0,1)-0.6180E+01   (1,0,1) 0.1565E+02   (1,0,1)-0.2359E+01   (1,0,1) 0.1999E+00
 (0,1,1)-0.1565E+02   (0,1,1)-0.6180E+01   (0,1,1) 0.0000E+00   (0,1,1) 0.4786E+00
 (3,0,0)-0.3645E+03   (3,0,0)-0.6774E+03   (3,0,0) 0.0000E+00   (3,0,0) 0.0000E+00
 (2,1,0) 0.5789E+03   (2,1,0)-0.4032E+03   (2,1,0) 0.0000E+00   (2,1,0) 0.0000E+00
 (1,2,0)-0.2106E+03   (1,2,0)-0.8687E+03   (1,2,0) 0.0000E+00   (1,2,0) 0.0000E+00
 (0,3,0) 0.6595E+03   (0,3,0)-0.1487E+03   (0,3,0) 0.0000E+00   (0,3,0) 0.0000E+00
 (2,0,1) 0.3420E+03   (2,0,1)-0.5973E+02   (2,0,1) 0.0000E+00   (2,0,1) 0.0000E+00
 (1,1,1)-0.1583E+03   (1,1,1)-0.3292E+03   (1,1,1) 0.0000E+00   (1,1,1) 0.0000E+00
 (0,2,1)-0.2806E+03   (0,2,1) 0.1574E+03   (0,2,1) 0.0000E+00   (0,2,1) 0.0000E+00
 (1,0,2) 0.1463E+03   (1,0,2) 0.1466E+02   (1,0,2) 0.3928E+01   (1,0,2) 0.6816E+00
 (0,1,2)-0.1466E+02   (0,1,2) 0.1463E+03   (0,1,2) 0.0000E+00   (0,1,2)-0.2885E+00

Beta as a function of delta         Gamma as a function of delta      Alpha as a function of delta
 (0,0,0) 0.4928574830467E+01        (0,0,0) 0.1988486850314E+01     (0,0,0)-0.2966547865984E+01
 (0,0,1)-0.1047324406285E+02        (0,0,1) 0.9654272578647E+00     (0,0,1) 0.2708152428548E+01
 (0,0,2) 0.2300465207896E+02        (0,0,2) 0.1830740292555E+01     (0,0,2)-0.6290500860821E+01

  beta*gamma-alpha**2 as a function of delta
  (0,0,0) 0.1000000000000E+01
```

The Numerical Results

The one-turn map around the δ-dependent orbit and fully normalised

$$A_{NL} = \exp\left(F \cdot \nabla\right) I = exp\left(F_2 \cdot \nabla + F_3 \cdot \nabla + \cdots\right) I$$

M=NORMAL%A_NL**(-1)*M*NORMAL%A_NL and NORMAL%A_NL

(1,0,0) -0.9301E+00	(1,0,0) -0.3674E+00	(1,0,0) 0.1000E+01	(1,0,0) 0.0000E+00
(0,1,0) 0.3674E+00	(0,1,0) -0.9301E+00	(0,1,0) 0.0000E+00	(0,1,0) 0.1000E+01
(2,0,0) 0.0000E+00	(2,0,0) 0.0000E+00	(2,0,0) -0.1334E+02	(2,0,0) -0.2024E+01
(1,1,0) 0.0000E+00	(1,1,0) 0.0000E+00	(1,1,0) -0.1297E+02	(1,1,0) 0.2669E+02
(0,2,0) 0.0000E+00	(0,2,0) 0.0000E+00	(0,2,0) 0.8940E+01	(0,2,0) 0.6486E+01
(1,0,1) -0.6180E+01	(1,0,1) 0.1565E+02	(1,0,1) 0.0000E+00	(1,0,1) 0.0000E+00
(0,1,1) -0.1565E+02	(0,1,1) -0.6180E+01	(0,1,1) 0.0000E+00	(0,1,1) 0.0000E+00
(3,0,0) 0.3175E+03	(3,0,0) -0.8038E+03	(3,0,0) 0.2085E+03	(3,0,0) -0.1005E+03
(2,1,0) 0.8038E+03	(2,1,0) 0.3175E+03	(2,1,0) 0.2896E+02	(2,1,0) 0.1393E+03
(1,2,0) 0.3175E+03	(1,2,0) -0.8038E+03	(1,2,0) 0.1088E+02	(1,2,0) 0.1080E+03
(0,3,0) 0.8038E+03	(0,3,0) 0.3175E+03	(0,3,0) -0.7418E+02	(0,3,0) 0.2115E+03
(2,0,1) 0.0000E+00	(2,0,1) 0.0000E+00	(2,0,1) 0.3014E+03	(2,0,1) 0.1596E+03
(1,1,1) 0.0000E+00	(1,1,1) 0.0000E+00	(1,1,1) 0.3216E+03	(1,1,1) -0.6029E+03
(0,2,1) 0.0000E+00	(0,2,1) 0.0000E+00	(0,2,1) -0.2919E+03	(0,2,1) -0.1608E+03
(1,0,2) 0.1463E+03	(1,0,2) 0.1466E+02	(1,0,2) 0.0000E+00	(1,0,2) 0.0000E+00
(0,1,2) -0.1466E+02	(0,1,2) 0.1463E+03	(0,1,2) 0.0000E+00	(0,1,2) 0.0000E+00

The Numerical Results
The tune and the normalised map

$$\mu = 2\pi\nu = 0.440 + 2.68\delta - 6.39\delta^2 - 138\left(z_1^2 + z_2^2\right)$$

```
TOTAL TUNE
   (0,0,0) 0.4401316125550E+00
   (0,0,1) 0.2677355781512E+01
   (2,0,0)-0.1375407883624E+03
   (0,2,0)-0.1375407883624E+03
   (0,0,2)-0.6386077990913E+01

   Comparing normalised map and cos(mu)x+sin(mu)p
   (1,0,0)-0.9301E+00    (1,0,0)-0.9301E+00
   (0,1,0) 0.3674E+00    (0,1,0) 0.3674E+00
   (1,0,1)-0.6180E+01    (1,0,1)-0.6180E+01
   (0,1,1)-0.1565E+02    (0,1,1)-0.1565E+02
   (3,0,0) 0.3175E+03    (3,0,0) 0.3175E+03
   (2,1,0) 0.8038E+03    (2,1,0) 0.8038E+03
   (1,2,0) 0.3175E+03    (1,2,0) 0.3175E+03
   (0,3,0) 0.8038E+03    (0,3,0) 0.8038E+03
   (1,0,2) 0.1463E+03    (1,0,2) 0.1463E+03
   (0,1,2)-0.1466E+02    (0,1,2)-0.1466E+02
```

Some General Results

$< x^2 >$ time average for example

This result depends only on the invariants! Why?

```
Average of x2 around (0,0,0)
   (1,1,0) 0.2464287415234E+01
   (0,0,2) 0.7625536895708E+00
   (1,1,1)-0.1377491138573E+02
   (0,0,3)-0.1898861439002E+01
   (2,2,0) 0.1217605130043E+04
   (1,1,2) 0.4965782620593E+02
   (0,0,4) 0.4409284770470E+01

Average of x2 around eta(delta)
   (1,1,0) 0.2464287415234E+01
   (1,1,1)-0.5236622031423E+01
   (2,2,0) 0.1217605130043E+04
   (1,1,2) 0.1150232603948E+02
```

Let us talk about this calculation which is really central to an understanding of perturbation theory.

Some General Results

$< x^2 >$: time average for example

Let us write a time average of a function F, for example, $F = x^2$.

$$\langle F \rangle = \frac{F + F \circ m + F \circ m \circ m + \cdots + F \circ m^N}{N} \quad N \to \infty \tag{1}$$

Using normal form to do the average!

Using normal form for average

$$m = a \circ r \circ a^{-1} \tag{2}$$

$$\begin{aligned}
\langle F \rangle &= \frac{F \circ a \circ a^{-1} + F \circ a \circ r \circ a^{-1} + \cdots + F \circ \left\{ a \circ r \circ a^{-1} \right\}^N}{N} \quad N \to \infty \\
&= \frac{F \circ a + F \circ a \circ r + \cdots + F \circ a \circ r^N}{N} \circ a^{-1} \quad N \to \infty \\
&= \frac{\overline{F} + \overline{F} \circ r + \cdots + \overline{F} \circ r^N}{N} \circ a^{-1} \; N \to \infty
\end{aligned} \tag{3}$$

Using normal form to do the average by going in phasors' basis!

Using normal form for average

Phasors' basis

$$\Lambda = \begin{pmatrix} e^{-i\mu} & 0 \\ 0 & e^{i\mu} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}}_{c^{-1}} \underbrace{\begin{pmatrix} \cos(\mu) & \sin(\mu) \\ -\sin(\mu) & \cos(\mu) \end{pmatrix}}_{r} \underbrace{\begin{pmatrix} 1/2 & 1/2 \\ -i/2 & i/2 \end{pmatrix}}_{c}$$

$$\begin{aligned} \overline{F} \circ r^k \circ c &= \underbrace{\overline{F} \circ c}_{F^r} \circ c^{-1} \circ r^k \circ c \\ &= \underbrace{\overline{F} \circ c}_{F^r} \circ \underbrace{\left\{c^{-1} \circ r \circ c\right\}}_{\Lambda}{}^{k} \\ &= F^r \circ \Lambda^k \end{aligned}$$

Expand F^r

Phasors' basis

$$z_1 = x + ip_x \qquad z_2 = x - ip_x \Longrightarrow z_1 z_2 = x^2 + p_x^2 = 2J$$

$$F^r = \sum F^r_{mn} z_1^m z_2^n$$

$$F^r \circ \Lambda^k = \sum F^r_{mn} z_1^m z_2^n \exp\left(i\mu\left(z_1 z_2\right)(n-m)\right)$$

$$\langle F^r \rangle = \sum F^r_{nn} z_1^n z_2^n$$

How is it done in the main programme?

Calling library routines

```
X=FIX(1)+ DX_1 ;
CALL AVERAGE(X,A_CS,X_AVERAGE)
WRITE(6,*) " NON-LINEAR DISPERSION DEFINED AS <X> = D<X>/DJ * J +  ..."
WRITE(6,*) " TAYLOR MAP RESULTS FOR <X> = D<X>/DJ * J +  ..."
CALL PRINT(X_AVERAGE,TITLE=" <X> FROM MAP NORMAL FORM")
ID=1
ID%V(1)=2.0_DP*ID%V(1)
X_AVERAGE=X_AVERAGE*ID
CALL PRINT(X_AVERAGE,TITLE=" <X> FROM MAP NORMAL FORM IN TERMS OF J")
```

$\longleftarrow \langle F^r \rangle = \sum F^r_{nn} x_1^n x_2^n$ (annotating `CALL AVERAGE(X,A_CS,X_AVERAGE)`)

$\longleftarrow \langle F^r \rangle = \sum F^r_{nn} 2^n J_1^n$ (annotating `ID%V(1)=2.0_DP*ID%V(1)` / `X_AVERAGE=X_AVERAGE*ID`)

How is it done in the library?

You could program yourself!

```
   SUBROUTINE AVERAGE(F,A_CS,F_FLOQUET)
    IMPLICIT NONE
    TYPE(MY_MAP) A_CS
    TYPE(MY_TAYLOR) F,F_FLOQUET
    INTEGER I

! 1) CREATES THE CHANGE OF BASIS FROM A ROTATION
!   TO A FULLY DIAGONAL COMPLEX MATRIX
    CALL CREATE_PHASORS()

!2) APPLIES A TO THE FUNCTION AND
! THEN GOES INTO THE COMPLEX PHASORS BASIS
    F_FLOQUET=(F*A_CS)*C_PHASOR

    DO I=0,N_MONO
      IF(JORDER(I)>MY_ORDER) CYCLE
! 3) REJECTS TERMS OF UNEQUAL POWERS
      IF(JEXP1(I)/=JEXP2(I)) THEN
       F_FLOQUET%A(I)=0.0_DP
      ENDIF
    ENDDO

   END SUBROUTINE AVERAGE
```

CALL CREATE_PHASORS() $\longleftarrow \Lambda = \underbrace{\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}}_{c^{-1}} \underbrace{\begin{pmatrix} \cos(\mu) & \sin(\mu) \\ -\sin(\mu) & \cos(\mu) \end{pmatrix}}_{r} \underbrace{\begin{pmatrix} 1/2 & 1/2 \\ -i/2 & i/2 \end{pmatrix}}_{c}$

F_FLOQUET=(F*A_CS)*C_PHASOR $\longleftarrow F^r = F \circ A_{cs} \circ c = \sum F^r_{mn} z_1^m z_2^n$

F_FLOQUET%A(I)=0.0_DP $\longleftarrow \langle F^r \rangle = \sum F^r_{nn} (z_1 z_2)^2$