SOLARIS

NATIONAL SYNCHROTRON
RADIATION CENTRE

TRACY

User's Manual

September 1, 2014

Contents

1 Introduction

Tracy is a tracking code written in C/C++, boosted with GSL (GNU Scientific Library), used for beam dynamics calculations. Main idea standing behind Tracy is to provide scientific library to boos and simplify computation of such physical quantities as emittance, twiss parameters, closed orbit, track of particle from one point to another etc.

This manual is a supplement of Tracy-2 User's Manual [1] and the goal is to clarify concepts used in this library and its implementation in code.

2 Physical concepts

2.1 Hamiltonian formalism

To be able to track beam in external magnetic field, computational tool to simplify calculations is requested. *Hamiltonian formalism* is a common used method to solve equations of movement and find time evolution of system. Uppermost quantity is Hamiltonian and it is defined in phase space. Apart from the abstract mathematical details of this formalism, from experimental point of veiw, phase space is very convenient domain, because both position $\overrightarrow{r}$ and momentum $\overrightarrow{p}$ can be measured, therefore Hamiltonian can be estimated basing on experiment. Relativistic phase space is defined by vector: $(x, p_x, y, p_y, ct, p_t)$. For example, relativistic Hamiltonian of charged particle around a reference trajectory in external magnetic field is given by [1]:

$$H = -(1+h_{ref}x)\left[\frac{q}{p_0}A_s + \sqrt{1-\frac{2}{\beta}p_t+p_t^2-\left(p_x-\frac{q}{p_0}A_x\right)^2-\left(p_y-\frac{q}{p_0}A_y\right)^2}\right]-\frac{1}{\beta}p_t \quad (1)$$

where

$p_t = -\frac{E-E_0}{p_0 c}$, where E_0 is reference energy
h_{ref} — local curvature
p_0 — reference momentum,
q — electric charge,
$\overrightarrow{B} = \nabla \times \overrightarrow{A}$ — magnetic vector potential A reliance on magnetic field B.

Definine new variable $\delta = \frac{p-p_0}{p_0}$. Applying canonical transformation, ultra-relativistic limitation and Hamiltonian expansion to second order leads to the following Hamilton's equations:

$$
\begin{aligned}
x' &= \frac{\partial H}{\partial p_x} = \frac{p_x}{1+\delta} \\
p'_x &= -\frac{\partial H}{\partial x} = \frac{q}{p_0} B_y + h_B \delta - h_B^2 x \\
y' &= \frac{\partial H}{\partial p_y} = \frac{p_y}{1+\delta} \\
p'_y &= -\frac{\partial H}{\partial y} = -\frac{q}{p_0} B_x \\
-cT' &= \frac{\partial H}{\partial \delta} = h_B x
\end{aligned}
\tag{2}
$$

2.2 4×5 Matrix formalism

Hamiltonian expansion can be concerned as multipole expansion. It means, that magnetic field is represented as superposition of dipole, quadruple, octupole etc. As far as extended equations are linear, they can be solved separately with each component.

4×5 Matrix formalism is basic way to solve Hamilton's equations as multipole expansion around reference curve x_{ref} with phase space vector defined as $\overrightarrow{x} = (x, p_x, y, p_y, \delta)$. Solving equations comes down to matrix multiplication using Jacobian M of this transformation.

Nonetheless, this formalism has some limitations. Tensor equation given by:

$$
\bar{x}^f = M\bar{x}^i + \mathscr{O}(\bar{x}^2), \tag{3}
$$

is linear and symplectic up to second order in Taylor expansion, so higher order multipoles cannot be directly computed. Therefore *thin kicks* has been applied. Idea is to split linear part of magnets in two separate magnets with length $\frac{L}{2}$ each and fill the gap with infinitesimally thin higher order kick. Strength of this kicks must be preserved regardless of thickness. Integrating Hamilton's equations with Dirac delta functions for multipole extension produces the kicks. Figure 1 represents idea of thin kicks.

The goal is to concatenate all linear elements with dipole kicks and misalignments into 6×6 matrix.

2.3 Symplectic Integrator

Symplectic geometry is a natural consequence of Hamiltonian formulation defined in phase space. According to Gromov Non-squeezing theorem, every canonical transformation (such as Hamilton's equations) preserves total area in phase space.

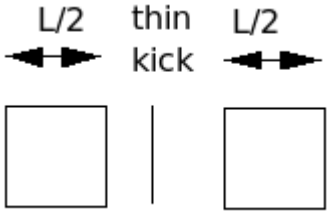

Figure 1: Magnet Model[1]

This conclusion has strong impact on long time stability for simulations, because even after many turns of computations, symplectic integration prevent from ending with strange structures as sinks, strange attractors, etc. Comparing with matrix based computations, as Forest-Ruth scheme, precision in symplectic integration is one order of magnitude higher with the same computational complexity [2].

If Hamiltonian can be spitted to two integrable parts,

$$H = H_1 + H_2 \tag{4}$$

symplectic map can be generated in following form:

$$e^{-LH} = e^{-\frac{L}{2}H_1} e^{-LH_2} e^{-\frac{L}{2}H_1} + \mathscr{O}(L^3) \tag{5}$$

Comparing with schematic in fig. 1, H_1 refers to drift space and H_2 is a thin kick.

For gaining precision, order of symplectic integrator can be incremented to 4. Then symplectic map can be presented as:

$$e^{-LH} = e^{-c_1 LH_1} e^{-d_1 LH_2} e^{-c_2 LH_1} e^{-d_2 LH_2} + e^{-c_2 LH_1} e^{-d_1 LH_2} e^{-c_1 LH_1} + \mathscr{O}(L^5) \tag{6}$$

which corresponds 4 drift spaces and 3 thin kicks placed alternately.

Values of coefficients c_1, c_2, d_1, d_2 are:

$$\begin{aligned} c_1 &= \frac{1}{2(2-2^{1/3})} \\ c_2 &= \frac{1-2^{1/3}}{2(2-2^{1/3})} \\ d_1 &= \frac{1}{2-2^{1/3}} \\ d_2 &= -\frac{2^{1/3}}{2-2^{1/3}} \end{aligned} \tag{7}$$

For more detailed informations about symplectic integration please refer to [2] and [3].

2.4 Magnet errors

As far as magnetic field can be well predicted and described inside magnet, field on the edges has more complex structure. In 4×5 Matrix formalism, we have to provide reference trajectory around which field can be expand, therefore it cannot be applied for dynamical magnet error calculations. Only linear lattice design with no errors can use matrix formalism.

Symplectic integrator and automatic differentiation allows to compute non-linear maps, so self-correcting model around perturbed reference trajectory can be implemented.

2.5 Closed Orbit finding

To prevent excessive beam loss, closed orbit calculations should be done properly. It can be numerically found by applying some perturbations of designed orbit as misalignments and tilt errors of magnets. For each iteration, one turn map has to be computed. One turn map is linear. To get this matrix in 4×5 matrix formalism, linear transfer matrices of every magnet should be concatenated. In symplectic integrator routines, non-linear map is expanded to desired order. Linear map requires reference trajectory. In circular cases (as ring orbit computation) closed orbit should be used as reference.

Closed orbit correction is applied by local bump method. Idea is to recursive scaling of bump angles named θ_1, θ_2, θ_3 to fulfill limitations. For more detailed description please refer to chapter 4 in [4].

3 Software implementation

In most cases, input data set for Tracy should be provided with Lattice file. It allows automatically setting all parameters needed in further computations. As well as reading lattices, output data from Tracy can be stored in other lattices. An example script presenting some basic usage of Tracy is shown in listing 1.

```
1  #define NO 1
2
3  #include <tracy_lib.h>
4  #include <stdio.h>
5
6  int no_tps = NO; // arbitrary TPSA order
7
8  extern bool  freq_map;// dynamic aperture
9  const double  delta   = 4.5e-2; // delta for off-momentum
       aperture
10
```

```
int main(int argc, char *argv[])
{

  long int lastpos;
  const long  seed = 1121;

  // Init random number generator
  iniranf(seed); setrancut(2.0);

 //globval initial values
  globval.RingType     = 1;
  globval.bpm          = 0;
  globval.Aperture_on = true;


  /* ***************************
   * read lattice and set globval
   * ***************************/

  char *lattice_filename = argv[1];
  Read_Lattice(lattice_filename);

  // force matrix computation
  globval.MatMeth     = true;

  /* ***************************
   * Do simulation / computations
   * ***************************/

  Ring_GetTwiss(true, 0e-2); // compute Twiss one-turn matrix
      //w or w/o chromaticities and energy offset

  printglob(); // print globalval and one-turn matrix

  prt_lat("lat.out", globval.bpm, true); // print to file

  getcod(0.0,lastpos); // compute closed orbit
  prt_cod("cod.out",ElemIndex("bpm_m"),true); //print to file

}
```

Listing 1: Tracy example

Every code should begin with arbitrary defined Truncated Power Series Algebra (TPSA) order equals 1. No other order value is supported.

```
int no_tps = 1;
```

Some simulations may require random number generator, therefore it can be initialized by setting seed and random cut value with commands:

```
18 iniranf(seed_value);
19 setrancut(cut_value);
```

User can also define some global values regarding to current requirements. All parameters are stored in structure `globvalrec`. Names, types and short description of each component are presented in table 1.

Table 1: globvalrec structure

type	**name**	**description**
double	dPcommon	dp for numerical differentiation
	dPparticle	energy deviation
double	delta_RF	RF acceptance
Vector2	TotalTune	transverse tunes
double	Omega	
	U0	energy lost per turn in keV
	Alphac	alphap
Vector2	Chrom	chromaticities
double	Energy	ring energy
long	Cell_nLoc	number of elements
	Elem_nFam	number of families
	CODimax	maximum number of cod search before failing
double	CODeps	precision for closed orbit finder
Vector	CODvect	closed orbit
int	bpm	bpm number
int	hcorr	horizontal corrector number
int	vcorr	vertical corrector number
int	qt	vertical corrector number
int	gs	girder start marker
int	ge	girder end marker
Matrix	OneTurnMat	oneturn matrix
	Ascr	
	Ascrinv	
	Vr	real part of the eigenvectors
	Vi	imaginal par of the eigenvectors
bool	MatMeth	matrix method
	Cavity_on	if true, cavity turned on

	radiation	if true, radiation turned on
	emittance	if true, emittance turned on
	quad_fringe	dipole- and quadrupole hard-edge fringe fields
	H_exact	”small ring” Hamiltonian.
	pathlength	absolute path length
	stable	
	Aperture_on	if true, aperture turned on
	EPU	Elliptically Polarizing Undulator
	wake_on	
double	dE	energy loss
	alpha_rad[DOF]	damping coeffs.
	D_rad[DOF]	diffusion coeffs (Floquet space)
	J[DOF]	partition numbers
	tau[DOF]	damping times
bool	IBS	intrabeam scattering
double	Qb	bunch charge
	D_IBS[DOF]	diffusion matrix (Floquet space)
Vector	wr, wi	real and imaginary part of eigenvalues
Vector3	eps	3 motion invariants
	epsp	transverse and longitudinal projected emittances
int	RingType	1 if a ring (0 if transfer line)

It is recommended to configure desired parameters before making any computation, because some values determinate software behaviour (f.ex. RingType parameter affect lattice file parsing into cell structure, MatMeth determine if matrix or symplectic formalism is implemented). Lines 20–23 in listing 1 shows how to set this values.

3.1 Lattice file parsing

After all initialization commands, user can parse lattice file using function: `void Read_Lattice(char *)`. Cell structure is then initialized and proper values from lattice file are stored there. During reading lattice also `globvalrec` structure is affected. Depending on `RingType` parameter, following values are applied:

Table 2: globvalrec default values

RingType=1	RingType=0
H_exact = false	Cavity_on = false
quad_fringe = false	radiation = false
EPU = false	emittance = false
Cavity_on = false	pathlength = false
radiation = false	CODimax = 10
IBS = false	dPparticle = dP[1]
emittance = false	
pathlength = false	
CODimax = 40	
delta_RF = 6%+ε[2]	
dPparticle = dP[1]	

If user want to change default global parameters, it should be done after lattice file parsing. It is not recommended to change non-configuration parameters (as dPparticle, CODvect, OneTurnMat etc.) if not necessary.

3.2 MatMeth dependencies

One of the most essential parameter determining if matrix formalism or symplectic integrators are applied during calculations, is `MatMeth`. Matrices are used if value is true, symplectic integrators otherwise.

In general, this parameter is set during parsing lattice file. Value of `Method` flag can be set to 0 for matrix formalism, 2 for 2^{nd} order symplectic integrator or 4 for 4^{th} order of integrator. However, 2^{nd} order integrator is not implemented and should not be used. `MatMeth` flag is set regarding `Method` flag value.

One thing that need to be pointed out is forcing changing the method. If `Method` is equal to 0, setting `MatMeth` to false ends up with series of "singular map" warnings and results are not accurate. On the contrary, if `Method` is equal to 4, calculations can be forced to matrix formalism without losing accuracy. Therefore it is recommended to keep `Method` equal 4 in lattice.

3.3 Physical computations

Taking into consideration an example shown in listing 1, whole configuration is done up to 34^{th} line. To make some calculations, user need only to call proper

[1] Value is calculated from lattice data

[2] arbitrary set to 0.060001 that is 6% + energy acceptance used in Soleil

function. For example, Twiss parameters can be reached by calling function `Ring_GetTwiss`, closed orbit can be calculated by calling `getcod` function etc.

Results of this kind of functions are stored in structures like `globvalrec` and can be written in file by calling one of the following function:

- void prt_lat(const char *fname, const int Fnum, const bool all) — user-defined output filename, Fnum determine which component characteristics should be written, all determine if only Fnum component should be written (false) or all data set
- void prt_chrom_lat(void) — output filename chromlat.out
- void prt_cod(const char *file_name, const int Fnum, const bool all) — same as prt_lat
- void prt_beampos(const char *file_name) — only beam position to be written
- void prt_codcor_lat(void) — output filename codcorlat.out
- void prt_beamsizes(void) — output filename beam_envelope.out

To print global variables and one-turn matrix on the screen, function `printglob()` should be used.

3.4 Example code output

Running previously explained example code, initialization should print similar output as following:

```
initializing TPSA library: no = 1, nv = 6, eps = 1.000000e-25

setrancut: cut set to 1.0

  TRACY III v. 3.5 compiled on Aug 21 2014

  LATTICE Statistics for today Thu Aug 28 16:25:10 2014

  Number of constants: UDIC                =   13, UDImax          =  100
  Number of keywords : nkw                 =   58, Lat_nkw_max     =  200
  Number of Families : globval.Elem_nFam   =   64, Elem_nFamMax    = 3000
  Max number of Kids : nKidMax             =  800, nKidMax         = 2000
  Number of Blocks   : NoB                 =   32, NoBmax          =  200
  Max Block size     : NoBE                = 7735, NoBEmax         =25000
  Number of Elements : globval.Cell_nLoc   = 6075, Cell_nLocmax    =25000
  Circumference      :  528.0000000 [m]

Lattice file: solaris-ring.lat
```

Depending on the MatMeth value, calculations time and one-turn matrix are different. Following output represent global values computed with symplectic integrators:

```
dPcommon     =  1.000e-06  dPparticle   =  0.000e+00  Energy [GeV] = 3.000
MaxAmplx [m] = -1.000e+01  MaxAmply [m] = -1.000e+01  RFAccept [%] = 6.00
MatMeth      =  FALSE      Cavity_On    =  FALSE      Radiation_On = FALSE
bpm          =    0        qt           =   0         Chambre_On   = FALSE
hcorr        =    0        vcorr        =   0

alphac       =   3.0731575109919619e-04
nux          =  42.1999806366295118      nuz  =  16.2800053918371326
ksix         =   1.024390                ksiz =   0.991448

OneTurn matrix:
matrix:
 1.081520e-01   8.941492e+00   0.000000e+00   0.000000e+00  -2.345941e-07   0.000000e+00
-1.056681e-01   5.101134e-01   0.000000e+00   0.000000e+00  -2.779522e-08   0.000000e+00
 0.000000e+00   0.000000e+00  -1.121566e+00   3.741313e+00   0.000000e+00   0.000000e+00
 0.000000e+00   0.000000e+00  -4.911417e-01   7.467369e-01   0.000000e+00   0.000000e+00
 0.000000e+00   0.000000e+00   0.000000e+00   0.000000e+00   1.000000e+00   0.000000e+00
-2.779522e-08  -1.288611e-07   0.000000e+00   0.000000e+00   1.622627e-01   1.000000e+00
```

Turning MatMeth flag to true, we obtain following output:

```
dPcommon     =  1.000e-06  dPparticle   =  0.000e+00  Energy [GeV] = 3.000
MaxAmplx [m] = -1.000e+01  MaxAmply [m] = -1.000e+01  RFAccept [%] = 6.00
MatMeth      =  TRUE       Cavity_On    =  FALSE      Radiation_On = FALSE
bpm          =    0        qt           =   0         Chambre_On   = FALSE
hcorr        =    0        vcorr        =   0

alphac       =   0.0000000000000000e+00
nux          =  42.1999966853987729      nuz  =  16.2800059677057334
ksix         =  37.464891                ksiz =  -8.916069

OneTurn matrix:
matrix:
 1.080483e-01   8.941734e+00   0.000000e+00   0.000000e+00  -2.384348e-07   0.000000e+00
-1.056722e-01   5.100253e-01   0.000000e+00   0.000000e+00  -2.824809e-08   0.000000e+00
 0.000000e+00   0.000000e+00  -1.121568e+00   3.741311e+00   0.000000e+00   0.000000e+00
 0.000000e+00   0.000000e+00  -4.911407e-01   7.467314e-01   0.000000e+00   0.000000e+00
 0.000000e+00   0.000000e+00   0.000000e+00   0.000000e+00   1.000000e+00   0.000000e+00
 0.000000e+00   0.000000e+00   0.000000e+00   0.000000e+00   0.000000e+00   1.000000e+00
```

As the example shows, one-turn matrix obtained using symplectic integrators contains more coefficients. However, time to proceed this is much longer than using matrices, therefore compromise between accuracy and time of computation has to be reached.

References

[1] J. Bengtsson, *Tracy-2 User's Manual*.

[2] L. Nadolski, J. LaskarLeslie Lamport, *Application of a new class of symplectic integrators to accelerator tracking*. IMCCE–CNRS, Paris, France, 2002. http://accelconf.web.cern.ch/AccelConf/e02/papers/weple076.pdf

[3] G. Parzen, *Symplectic Tracking Using Piont Magnets and a Reference Orbit Made of Circular Arcs and Straight Lines* , Bookhaven National Labolatory Associated Universities, Inc. Upton, NY 11973, June 1993 http://arxiv.org/pdf/physics/9811020v1.pdf

[4] A. Parfenova, G. Franchetti, B. Franczak, M. Kirk, C. Omet, *SIS18 closed orbit correction using a local bump method*, GSI, 64291 Darmstadt, Germany, November 10, 2006 http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=60770B0BE60D238196CD94BC46B7EF17?doi=10.1.1.161.8445&rep=rep1&type=pdf