Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

Beams
Department

Optimising Resonance Driving Terms Using MAD-NG Parametric Differential Maps.

I.FAST 9th Low Emittance Rings Workshop 2024

Laurent Deniau
CERN-BE/ABP

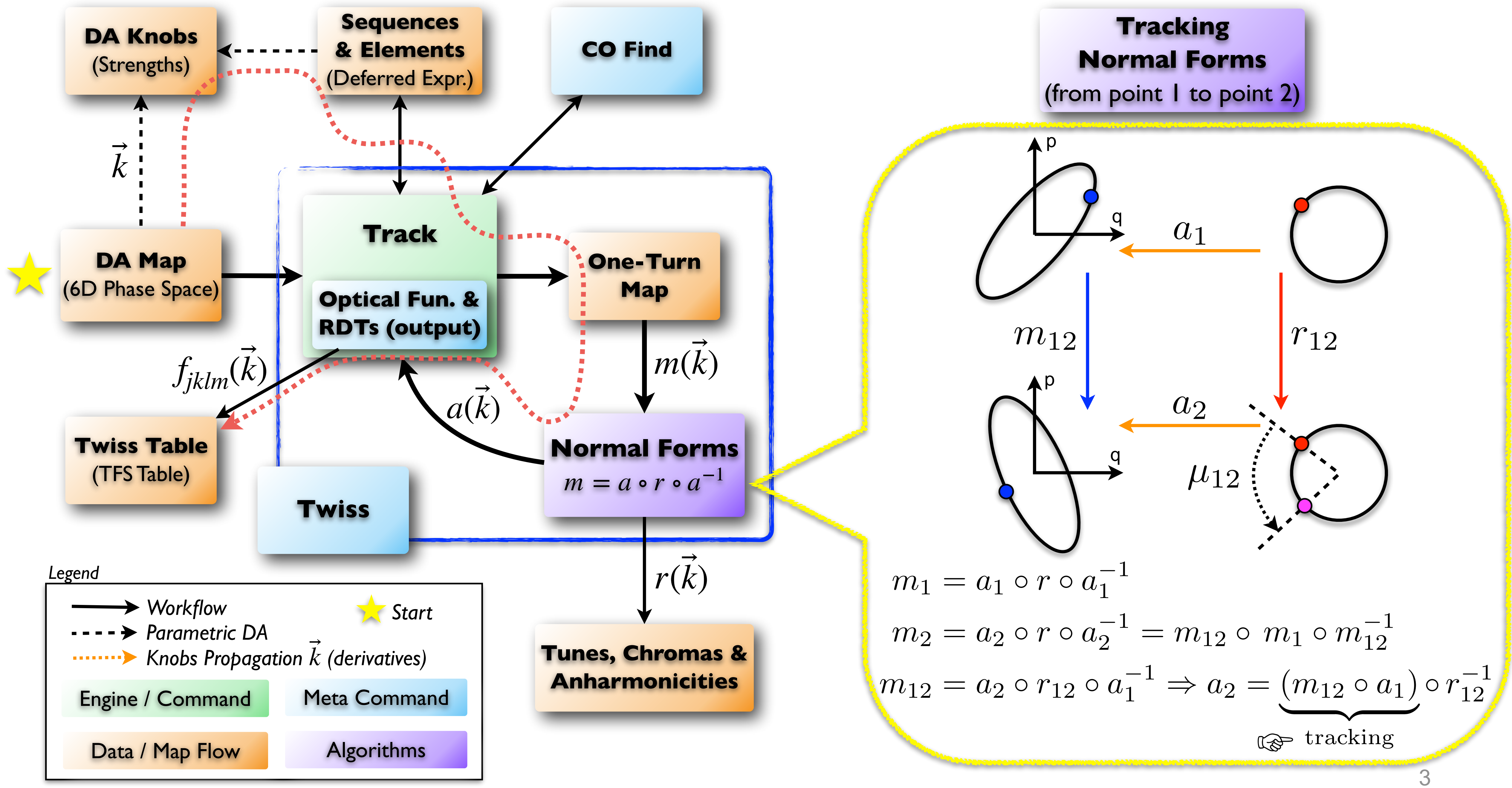
13-16 February 2024

MAD-NG's Features

- **Integrated platform designed for Methodical Accelerators Design (optics).**
  - ➡ **Flexible language** ➠ **fast, simple, and general purpose scripting language.**
    - ~70% of the code is written in the Lua(JIT) scripting language, ~30% in C and C++.
  - ➡ **Flexible technologies** ➠ **self-contained, all-in-one and modular application.**
    - Single "copy & run" application, no dependencies (requires Gnuplot installed for plotting).
  - ➡ **Efficient & Portable technologies** ➠ **embeds a Tracing Just in Time compiler.**
    - Same results everywhere (LNX, OSX, WIN), extensive unit tests (>8000) and examples.
    - Extremely simple and fast Foreign Function Interface to external libraries in C, C++, Fortran, etc…
  - ➡ **Easy to extend & support** ➠ **embeds an online profiler and debugger.**
    - Adding new elements with new physics takes a day (assuming known explicit exact equations).
- **5D and 6D physics using high-order differential algebra and symplectic integrators.**
  - ➡ Combined physics, combined elements, misalignments & errors, local & global frames, all elements fringe fields, forward, backward and reverse tracking.
  - ➡ Physics & Maths in Lua and C/C++, **performance is x10-x70 faster than MADX-PTC**.
  - ➡ Support and development for new physics extensions is extremely easy…
- **Development open source.**
  - ➡ License GPL V3, Manual (~200p, covers <25%), Lua Manual (30p).
  - ➡ Sources https://github.com/MethodicalAcceleratorDesign/MAD
  - ➡ Releases & Manual https://cern.ch/mad/releases/madng/
  - ➡ Online Manual https://cern.ch/mad/releases/madng/html/
  - ➡ Learn Lua in 15 min: https://www.youtube.com/watch?v=REReUFgii5A

MAD-NG's Parametric Differential Maps Principle

Track a **high-order differential algebra** (DA) map on the closed orbit (optionally) equipped with **parameters** (knobs) to obtain the **one-turn map** m, then compute the **closed non-linear normal form** $m = a \circ r \circ a^{-1}$ and track the normalising map a to extract the **optical functions** (α, β, μ, etc.) and the **resonant driving terms (RDTs)** along the lattice.



Laurent Deniau, CERN BE/ABP, 1211 Geneva, laurent.deniau@cern.ch

MAD-NG studies: RDTs for HL-LHC

BE Beams Department

```
-- HL-LHC setup
MADX:load("hllhc_saved.seq", "hllhc_saved.mad")
MADX.lhcb1.beam = beam {particle="proton", energy=450}
MADX.lhcb2.beam = beam {particle="proton", energy=450}
MADX.lhcb2.dir  = -1 -- bv = -1

-- list of RDTs
local rdts = {"f4000", "f3100", "f2020", "f1120"}

-- loop over lhcb1 and lhcb2
for _,lhc in ipairs{MADX.lhcb1, MADX.lhcb2} do

  -- create phase-space damap at 4th order
  local X0 = damap {nv=6, mo=4}

  -- compute RDTs along HL-LHC
  local mtbl = twiss {sequence=lhc, X0=X0, trkrdt=rdts}

  -- plot RDTs along HL-LHC
  plot_rdt(mtbl, rdts)

end -- end of loop
```

4th-order DA map

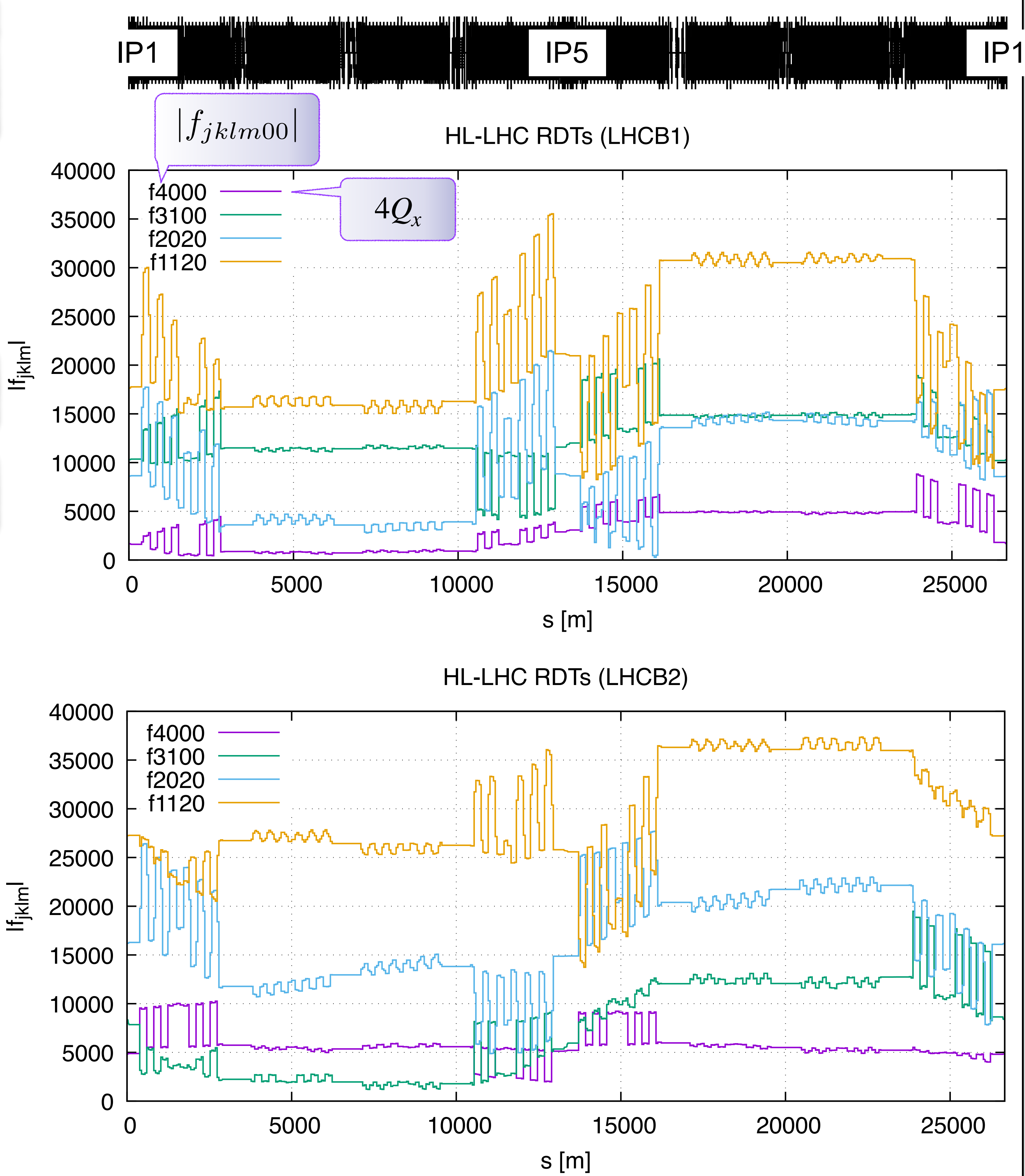
$$f_{jklm} = \frac{h_{jklm}}{1 - e^{2\pi i[(j-k)\nu_x + (l-m)\nu_y]}}$$

Resonances: $N = (j-k)Q_x + (l-m)Q_y$

Spectral lines: $H(1+k-j, m-l)$

$V(k-j, 1+m-l)$

These 2 RDTs' plots take 32 sec in MAD-NG 40 min in MADX-PTC



Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

MAD-NG studies: Parametric RDTs for HL-LHC

BE Beams Department

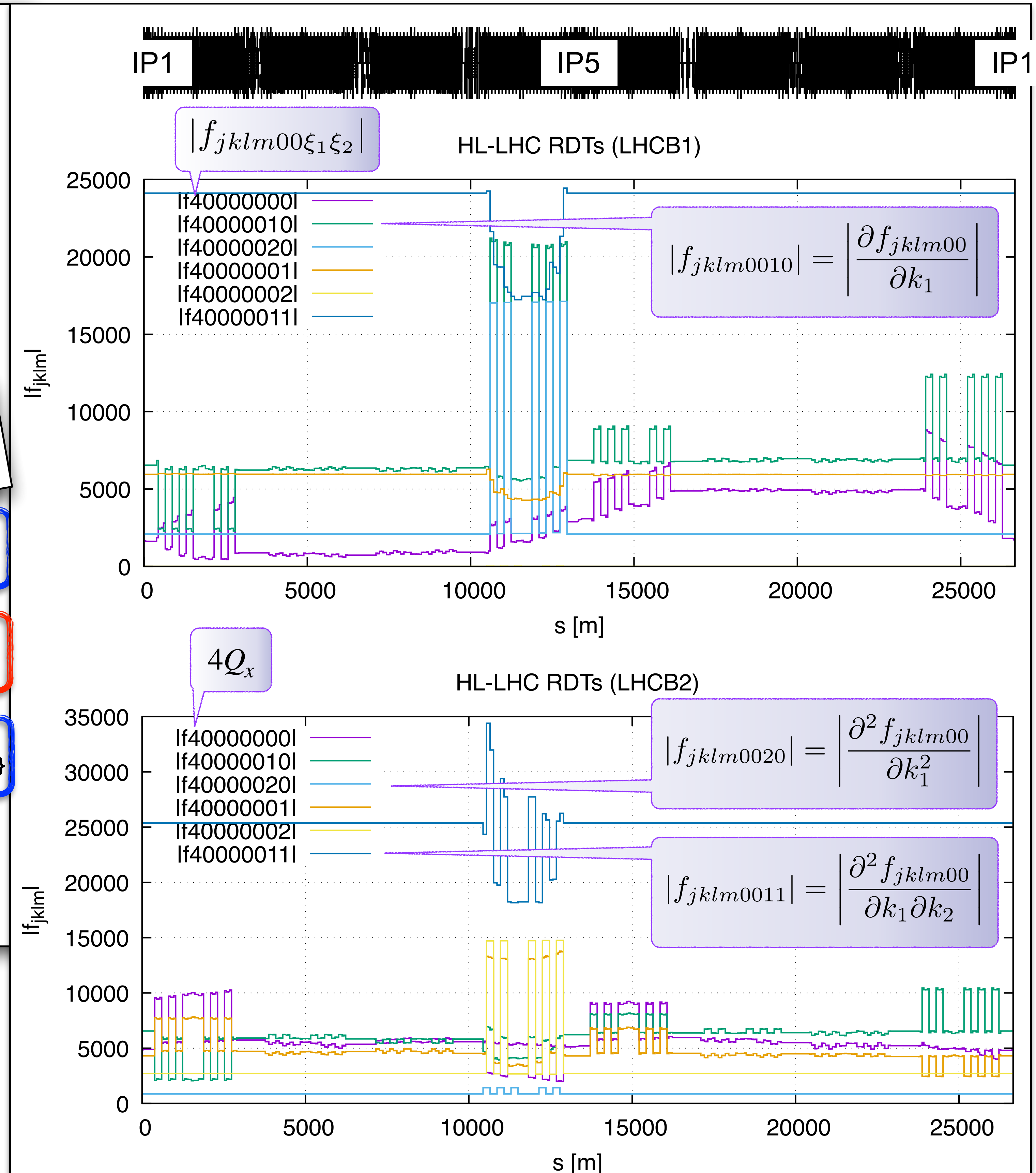
```
-- HL-LHC setup
MADX:load("hllhc_saved.seq", "hllhc_saved.mad")
MADX.lhcb1.beam = beam {particle="proton", energy=450}
MADX.lhcb2.beam = beam {particle="proton", energy=450}
MADX.lhcb2.dir  = -1 -- bv = -1

-- list of knobs for both sequences
local knbs = { LHCB1 = {'ksf1.a45b1','ksf2.a45b1'},
               LHCB2 = {'ksf1.a45b2','ksf2.a45b2'} }

-- list of RDTs: 4Qx w 1st and 2nd derivatives vs knobs
local rdts = {"f40000000", "f40000010", "f40000020",
              "f40000011", "f40000001", "f40000002",}

-- loop over lhcb1 and lhcb2
for _,lhc in ipairs{MADX.lhcb1, MADX.lhcb2} do

  -- select knobs
  local knb = knbs[lhc.name]

  -- create phase-space damap at 6th order (mo=4th+po)
  local X0 = damap {nv=6, mo=6, np=#knb, po=2, pn=knb}

  -- set knobs: scalar + TPSA -> TPSA
  for _,k in ipairs(knb) do MADX[k] = MADX[k]+X0[k] end

  -- compute RDTs along HL-LHC
  local mtbl = twiss {sequence=lhc, X0=X0, trkrdt=rdts }

  -- plot RDTs along HL-LHC
  plot_rdt(mtbl, rdts)

end -- end of loop
```

k_1 k_2

6th-order parametric DA map

$$f_{jklmno\xi_1\xi_2} = \frac{\partial^{\xi_1+\xi_2} f_{jklmno}}{\partial k_1^{\xi_1} \partial k_2^{\xi_2}}$$

$$\text{DA map order} = j+k+l+m+n+o+\sum_i \xi_i$$



Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

MAD-NG studies: lower RDTs for LHC@injection

Loading LHC Sequences & Optics (1)

```
MADX:load'lhc_seq.madx'
MADX:load'inj_optics.madx'
MADX.lhcb1.beam = beam {particle='proton', energy=450}
MADX.lhcb2.beam = beam {particle='proton', energy=450}
MADX.lhcb2.dir  = -1 -- set LHCB2 as reversed
```

Building Parametric DA Map (2)

```
local prms = { -- param./knob names (strings)
  -- 16 strengths of trim quadrupoles families
  'kqtf.a12b1', 'kqtf.a23b1', ···, 'kqtf.a81b1',
  'kqtd.a12b1', 'kqtd.a23b1', ···, 'kqtd.a81b1',
  -- 16 strengths of octupoles families
  'kof.a12b1' , 'kof.a23b1' , ···, 'kof.a81b1',
  'kod.a12b1' , 'kod.a23b1' , ···, 'kod.a81b1',
}
```

32 Circuit Knobs 16 MQT + 16 MO

```
-- DA map representing parametric phase-space
local X0 = damap {nv=6, mo=5, np=#prms, po=1, pn=prms}
```

5th-order parametric DA map

```
-- convert scalars to GTPSAs within MADX env.
for _,knb in pairs(prms) do
  MADX[knb] = MADX[knb] + X0[knb]
end
```

Use strengths as DA map parameters

Parametric Normal Forms & Setup (3)

```
-- function to compute non-linear normal forms
local function get_nf (lhc, X0)
  local _, mflw = track {sequence=lhc, X0=X0}
  return normal(mflw[1]):analyse("all")
end
```

Twiss-like RDTs @ IP1 (faster for single point)

```
-- save reference values
local nf = get_nf(X0, MADX.lhcb1)
local q1ref  = nf:q1{1}
local q2ref  = nf:q2{1}
local q1jref = nf:anhx{1,0}
local q2jref = nf:anhy{0,1}
```

A solution is found by: MAD-NG in 3 min MADX-PTC in 45 min (using finite difference approx.)

Optimizing RDTs (4 & 5)

```
match {
  -- compute non-linear normal forms
  command := get_nf(), -- returns nf used below
```

Jacobian [10x32] filled derivatives vs. knobs used by optimiser

```
  -- compute Jacobian from parametric maps
  jacobian = \nf,_,J =>
    for k=1,32 do -- fill [10x32] J matrix
      J:set(1,k, nf:q1{1,k} or 0)
      J:set(2,k, nf:q2{1,k} or 0)
      J:set(3,k, nf:anhx{1,0,0,k})
      J:set(4,k, nf:anhy{0,1,0,k})
      J:set(5,k, nf:gnfu{"2002",k}.re)
      J:set(6,k, nf:gnfu{"2002",k}.im)
      J:set(7,k, nf:gnfu{"4000",k}.re)
      J:set(8,k, nf:gnfu{"4000",k}.im)
      J:set(9,k, nf:gnfu{"0040",k}.re)
      J:set(10,k,nf:gnfu{"0040",k}.im)
    end
  end,
```

$$\frac{\partial Q_x}{\partial k_i}, \frac{\partial Q_y}{\partial k_i}, \frac{\partial^2 Q_x}{\partial J_x \partial k_i}, \frac{\partial^2 Q_y}{\partial J_y \partial k_i}$$

$$\frac{\partial f_{2002}}{\partial k_i}, \frac{\partial f_{4000}}{\partial k_i}, \frac{\partial f_{0040}}{\partial k_i}$$

```
  -- variables in MADX env. to use as knobs
  variables = {
    {name=prms[1] , var='MADX[prms[1]]' },
    ···,
    {name=prms[32], var='MADX[prms[32]]'},
  },
```

32 knobs to vary

10 constraints to satisfy

```
  -- target constraints as equalities to zero
  equalities = {
    {name = 'q1'    , expr = \nf -> nf:q1{1} - q1ref},
    {name = 'q2'    , expr = \nf -> nf:q2{1} - q2ref},
    {name = 'q1j1'  , expr = \nf -> nf:anhx{1,0} - q1jre
    {name = 'q2j2'  , expr = \nf -> nf:anhy{0,1} - q2jre
    {name = 'f2002r', expr = \nf -> nf:gnfu{"2002"}.re  - 0},
    {name = 'f2002i', expr = \nf -> nf:gnfu{"2002"}.im) - 0},
    {name = 'f4000r', expr = \nf -> nf:gnfu{"4000"}.re) - 0},
    {name = 'f4000i', expr = \nf -> nf:gnfu{"4000"}.im) - 0},
    {name = 'f0040r', expr = \nf -> nf:gnfu{"0040"}.re) - 0},
    {name = 'f0040i', expr = \nf -> nf:gnfu{"0040"}.im) - 0},
  },
} -- close match
```

Invariant

$$Q_x, Q_y, \frac{\partial Q_x}{\partial J_x}, \frac{\partial Q_y}{\partial J_y}$$

Minimise

f_{2002}

f_{4000}

f_{0040}

Match has the same structure as in MAD-X with “use_macro”

Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

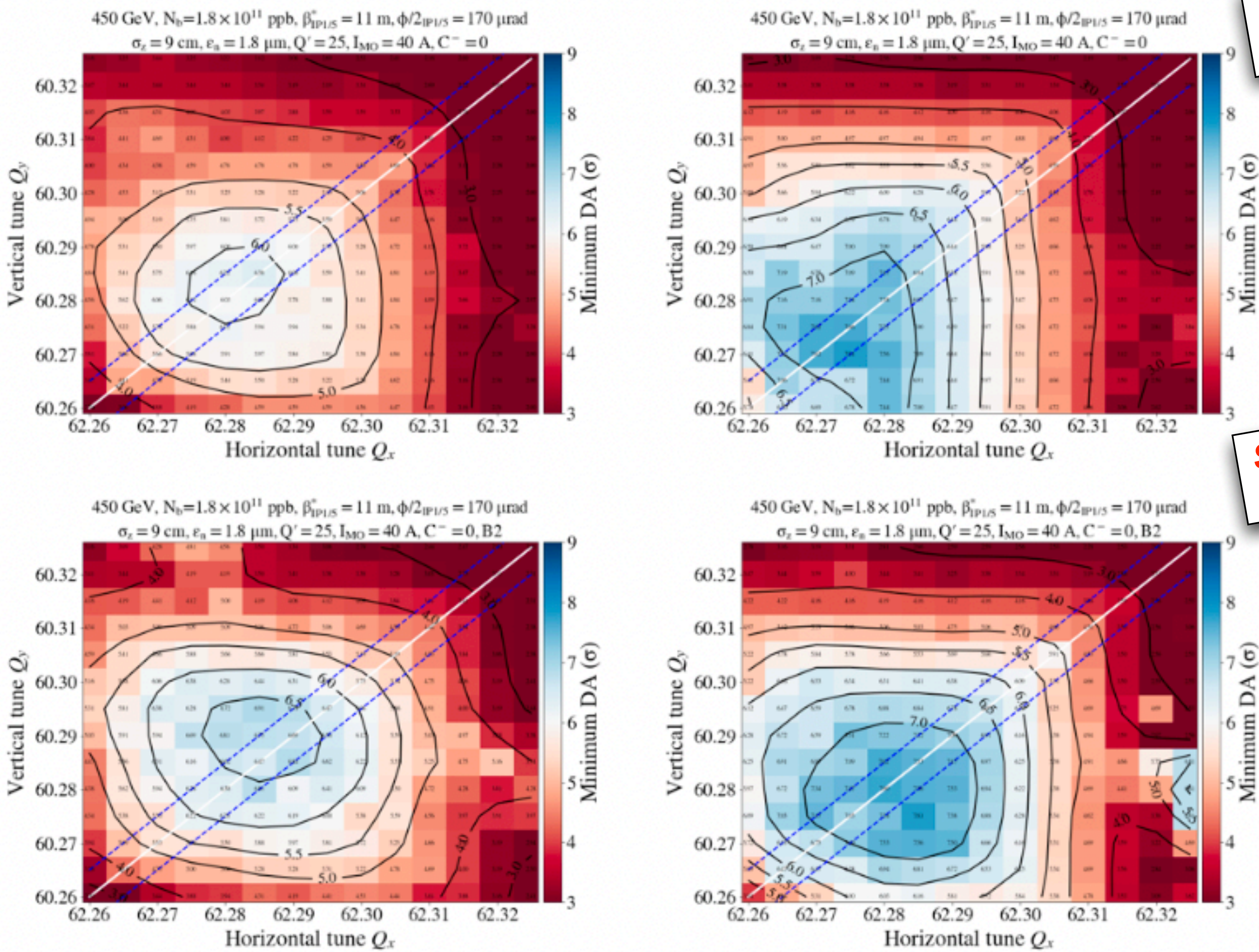
MAD-NG studies: DA increase for LHC@injection

BE Beams Department

Dynamic Aperture Improvements

Dynamic aperture for beam 1 (top) and beam 2 (bottom) with old (left) and new (right) injection optics for LHC. Lowering the octupolar RDTs has significantly improved the dynamic aperture at injection.

Courtesy Sofia Kostoglou



See Rogelio's paper for details

Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

References

- HB2023 papers
  - L. Deniau et al., "Optimising Resonance Driving Terms Using MAD-NG Parametric Differential Maps".
  - R. Tomás et al., "Optics for Landau Damping with Minimised Octupolar Resonances in the LHC".
  - K. Paraschou et al.,"Emittance Growth from Electron Clouds Forming in the LHC Arc Quadrupoles".

- Papers and books on Differential Algebra, Nonlinear Normal Forms and RDTs.

  - **E. Forest**, M. Berz and J. Irwin, "***Normal Form Methods for Complicated Periodic Systems using Differential Algebra and Lie Operators***", Particle Accelerators, Vol. 24, pp. 91-107, 1989.
  - E. Forest, "*From Tracking Code to Analysis, Generalised Courant-Snyder Theory for Any Accelerator Model*", Springer, 2016.
  - A. Franchi, "*Studies and Measurements of Linear Coupling and Nonlinearities in Hadron Circular Accelerators*", PhD Thesis, 2006.
  - R. Bartolini and F. Schmidt, "*Normal Form via Tracking or Beam Data*", Particle Accelerators, Vol. 59, pp. 93-106, 1998.
  - T. Pugnat, R. Tomás, A. Franchi and B. Dalena, "*Non-linear Variation of the Beta-beating Measured from Amplitude*", 12th Int. Particle Acc. Conf., 2021.
  - L. Deniau and C.I. Tomoiaga, "*Generalised Truncated Power Series Algebra for Fast Particle Accelerator Transport Maps*", 6th Int. Particle Acc. Conf., 2015.

Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

Extra Slides

MAD-NG's Physics I

- **5D and 6D physics using differential algebra and symplectic integrators.**
  - **combined physics & elements, slicing & frames, easy to extend, etc…**
  - **x10-30 faster than MADX-PTC for TPSA tracking (Multivariate Taylor Series).**
- **Survey**: geometrical tracking (*global frame, linear algebra*)
  - Survey supports **multi-turns**, **ranged** and step-by-step **forward, backward** and **reverse** tracking. Return a Survey table *and* a Survey map flow (tracked context).
  - fully compatible with Track for superposition and observable points (e.g. table output, smooth plots, slicing, actions, sub-elements, **local in global frame**, etc…)
  - support **exact** misalignments, **permanent** misalignments, and patches.
- **Track**: dynamical tracking (*local mobile frame, differential algebra*)
  - Track supports **multi-particles** or **multi-damaps**, **multi-turns**, **ranged** and step-by-step **forward**, **backward** and **reverse** tracking of **charged** particles to **arbitrary** DA **order** with an arbitrary number of **parameters** (few hundreds). Return a Track table *and* a Track map flow (tracked context).
  - fully compatible with Survey for superposition and observable points.
  - support **exact** misalignments, **permanent** misalignments, **multipoles** & field errors **for all elements.** Can be combined freely with **patches**.
  - **symplectic tracking with integrators up to 8th order** on 5D (delta-p) and 6D (delta-rf) phase space *(exact=true, time=true, totalpath e.g. for thick RF)*.
  - provides true **thick lens** and thin lens tracking model, **radiation with photons tracking** (disabled in twiss), **fringe fields** (hard edge for all elements, including quads, solenoid, RFs), **mutable particles** (multiple beams), **exact patches** (translations, rotations & time-energy), 4D weak-strong beam-beam (sixtracklib), apertures (all kinds).
  - can search for the closed orbit to support relative initial coordinates.

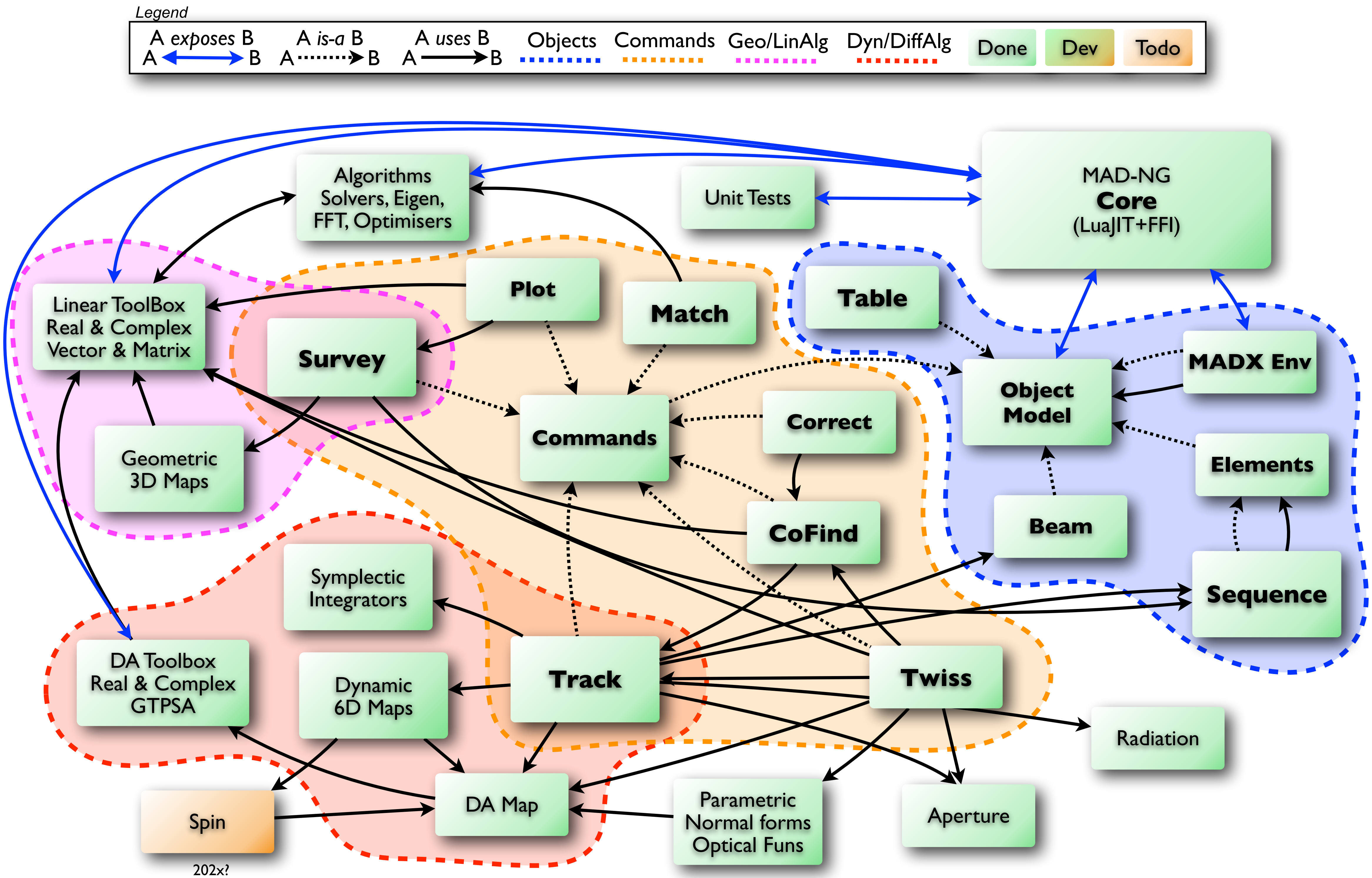
MAD-NG's Physics II

Beams Department

- **Cofind**: fix point search
  - Newton-based optimiser running **Track** with 1st order DA map or 7 particles (F-Diff).
  - extend Track with actions.
- **Twiss**: optics tracking
  - runs **Cofind** (closed orbit) - **Track** (one-turn map) - **Normal** - **Track** (optics, RDTs).
  - extend Track with actions to compute on-the-fly optics and fill twiss table (extended track table).
  - support strongly coupled optics, linear and non-linear dispersions, tunes, and chromaticities, RDTs, synchrotron integrals, compaction factor, phase slip factor, gamma transition, Montaigue functions, etc…
- **Match**: highly configurable optimiser
  - on the model of MAD-X `use_macro` approach, i.e. arbitrary user's setups & runs.
  - provides all kinds of local & global, linear & non-linear, optimiser (~20 algorithms).
  - very flexible, highly configurable with many physics-oriented setups (not just a penalty-function to minimise).
- **Correct**: orbit correction
  - provides few algorithms (e.g. SVD, Micado) to correct orbit using BPMs and Kickers. Supports many options.
- **Normal**: parametric normal forms & analysis
  - provides linear and non-linear parametric normal forms on DA map (used by twiss) to extract RDTs. Can also be applied at observable points in Track to track RDTs, either on-the-fly with actions or through post processing of DA maps saved in Track table.

Laurent Deniau, CERN BE/ABP, 1211 Geneva, laurent.deniau@cern.ch

MAD-NG's Ecosystem



GTPSA in a nutshell

Beams Department

- Generalised Truncated Power Series Algebra **IPAC 2015** **2017-2018** **Github MAD**
  - ➡ Multivariate Taylor polynomials of order n in ℝ & ℂ.
  - ➡ Powerful tool for solving differential equations (e.g. motion equations).

TPSA coefficients

1 variable x at order n in the *neighbourhood* of the point a in the domain of the function f:

$$T_f^n(x;a) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n = \sum_{k=0}^{n} \boxed{\frac{f_a^{(k)}}{k!}}(x-a)^k$$

convergence of the remainder (i.e. truncation error):

$$\lim_{n\to\infty} R_f^n(x;a) = \lim_{n\to\infty} f(x) - T_f^n(x;a) = 0$$

f(x) is an analytic function, $T_f^n(x;a)$ *is a polynomial approximation nearby* a *with radius of convergence h:* $\min_{h>0} \lim_{n\to\infty} R_f^n(a \pm h; a) \neq 0$.

2 variables *(x,y)* at order 2 nearby *(a,b)*:

$$= f^{(1)}_{(a,b)}(x-a, y-b)$$

$$T_f^2(x,y;a,b) = f(a,b) + \boxed{\left.\frac{\partial f}{\partial x}\right|_{(a,b)}(x-a) + \left.\frac{\partial f}{\partial y}\right|_{(a,b)}(y-b)} + \ldots$$

homogeneous polynomials

f must not depend on the integration path, i.e. must derive from a potential!

$$+\frac{1}{2!}\left(\boxed{\left.\frac{\partial^2 f}{\partial x^2}\right|_{(a,b)}(x-a)^2 + \boxed{2\frac{\partial^2 f}{\partial x \partial y}}\Bigg|_{(a,b)}(x-a)(y-b) + \left.\frac{\partial^2 f}{\partial y^2}\right|_{(a,b)}(y-b)^2}\right)$$

$$= f^{(2)}_{(a,b)}(x-a, y-b)$$

v variables X at order n nearby A: **TPSA coefficients**

$$T_f^n(X;A) = \sum_{k=0}^{n} \frac{f_A^{(k)}}{k!}(X;A)^k = \sum_{k=0}^{n} \boxed{\frac{1}{k!}} \sum_{\boxed{|\vec{m}|=k}} \boxed{\binom{k}{\vec{m}} \frac{\partial^k f}{\partial X^{\vec{m}}}}\Bigg|_A (X;A)^{\vec{m}} \quad \text{with} \quad \boxed{\binom{k}{\vec{m}}} = \frac{k!}{c_1!\,c_2!\ldots c_v!}$$

monomials of order k **multinomial**

Accuracy of TPSA (myths and legends)

- GTPSA are **exact** to machine precision, **no** approximation for orders 0..n
  - ➡ derivatives are computed using **automatic differentiation** (AD).

from Wikipedia

> AD exploits the fact that every computer program, no matter how complicated, executes a sequence of elementary arithmetic operations (addition, subtraction, multiplication, division, etc.) and elementary functions (exp, log, sin, cos, etc.). By applying the chain rule repeatedly to these operations, **derivatives of arbitrary order can be computed automatically, accurately to working precision**, and using at most a small constant factor more arithmetic operations than the original program.
>
> Symbolic differentiation can lead to **inefficient code** and faces the di[...] computer program into a single expression, while numerical differentiation can i[...] round-off errors in the discretization process and cancellation. **Both classical methods have problems with calculating higher derivatives, where complexity and errors increase.**

Functions of TPSAs ≠ TPSAs as functions
exact ≠ approximate

- MAD-NG includes a complete toolbox (i.e. module) to handle DA using AD…
  - ➡ users have full access to GTPSA and DAmaps from the scripting language.
  - ➡ users can manipulate DAmaps stored in the MTable or the MFlow returned by Track.
- *So when DAmap/TPSA introduce errors? (Something that we never do…)*
  - ➡ If they are used as *functions* (e.g. evaluated), instead of *DA* (e.g. track, twiss).
  - ➡ High orders of $T_f^n(x;a)$ are used to interpolate at the new position by substitution, e.g. MADX.

$$T_f^n(x;a+h) = \sum_{k=0}^{n} \frac{f_{a+h}^{(k)}}{k!}(x-a-h)^k \,; \quad f(a+h) \approx \underbrace{\sum_{k=0}^{n} \frac{f_a^{(k)}}{k!}h^k}_{T_f^n(a+h;a)} \,; \quad f_{a+h}^{(k)} \approx \frac{\mathrm{d}^k T_f^n(x;a)}{\mathrm{d}x^k}(a+h)$$

Matrix codes don't do better!

order n is constant
order n-1 is linear in h

Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch