SYMPLECTIC VERIFICATION OF TRACKING CODES USING TAYLOR SERIES METHODS

E. T. Navar Root[1], D. C. Sagan[1], M. G. Signorelli[1,2], G. H. Hoffstaetter[1,2]
[1]Cornell University, Ithaca, NY, USA
[2]Brookhaven National Laboratory, Upton, NY, USA

Abstract

Many accelerator tracking codes include interactions, models, and coordinate transformations that make it difficult to evaluate symplecticity analytically. The stability of long term tracking depends on preservation of Hamiltonian structure, making numerical verification of symplecticity important. We present a symplectic verification method based on truncated power series (TPS) propagation through polymorphic tracking codes. Initial phase-space coordinates are represented as TPS variables and passed directly through the tracking routine. The first-order TPS coefficients give the Jacobian matrix while higher-order coefficients retain the nonlinear structure of the map. Since each Jacobian element is itself a TPS, the symplectic condition $J^T SJ = S$ can be evaluated order-by-order, enabling verification of symplecticity up to the truncation order. We demonstrate the method on a weak-strong beam-beam simulation including hourglass and slingshot effects. These effects complicate direct analytic derivation of the full six-dimensional map, while TPS propagation verifies symplecticity directly through tracking.

INTRODUCTION

Symplecticity is a fundamental requirement for accelerator tracking derived from Hamiltonian motion. Small violations of symplectic structure may produce artificial emittance growth during long-term tracking. Modern beam dynamics simulations frequently combine exact maps, drift-kick integrators, slicing models, and coordinate transformations, making direct analytic verification of symplecticity increasingly difficult.

In practice, symplecticity can be assessed indirectly through long-term tracking observables such as dynamic aperture preservation or emittance conservation. While useful, these methods do not directly verify the implemented map and may fail to identify localized violations introduced by numerical approximations or implementation details.

Truncated power series algebra (TPSA) provides a direct alternative by propagating differentials through the tracking simulation itself. Instead of manually deriving the Jacobian or computing finite differences [1], the derivative map is generated algebraically through TPS propagation and stored directly in the truncated power series coefficients.

The method is particularly effective for polymorphic tracking codes. In a polymorphic framework, the same tracking routine can operate on arbitrary numerical types including floating point values, dual numbers, and truncated power series objects. The physics implementation therefore remains unchanged while the numerical type determines whether the simulation computes trajectories, derivatives, or higher-order maps. This makes TPS-based symplectic verification compatible with modern differentiable accelerator physics libraries like SciBmad [2].

In this paper, we describe a TPS-based symplectic verification framework and apply it to a weak-strong beam-beam simulation including longitudinal slicing, hourglass effects, and slingshot effects [3]. These effects produce a more precise collision model, but make direct analytic proof of symplecticity difficult.

POLYMORPHIC TRACKING

The tracking framework used in this work is fully polymorphic. All tracking routines are implemented generically over numerical type, allowing the same physics code to operate on floating point scalars, automatic differentiation types, or TPS objects such as those from `GTPSA.jl` [4] without modification.

This is achieved through type-generic element implementations in which all coordinates and intermediate quantities are promoted to a common numerical type before tracking. Under standard execution the promoted type resolves to `Float64`. When TPS variables are provided as inputs, all arithmetic operations instead propagate truncated power series coefficients through the full simulation automatically.

This design eliminates the need for separate derivative implementations. The same drift, beam-beam kick, coordinate transformation, and slicing routines used for tracking also generate the derivative map. Symplectic verification therefore operates on the implemented simulation rather than an independently derived approximation.

TPS SYMPLECTIC VERIFICATION

Initial canonical coordinates are represented as truncated power series expanded about a reference trajectory,

$$x = x_0 + \epsilon_x, \quad p_x = p_{x0} + \epsilon_{p_x} \ ,$$

with equivalent expressions for all six phase-space coordinates. These TPS variables are passed directly through the tracking simulation.

After propagation, each output coordinate becomes a multivariate truncated power series,

$$X_i = X_i^{(0)} + \sum_j \frac{\partial X_i}{\partial q_j} \epsilon_j + O(\epsilon^2) \ ,$$

where q_j denotes the initial canonical coordinates. The first-order TPS coefficients form the linear Jacobian matrix,

$$J_{ij} = \frac{\partial X_i}{\partial q_j} \,.$$

Unlike finite differences, the derivatives are propagated algebraically through the simulation itself and therefore do not suffer from step-size dependence or subtractive cancellation.

Because the map is represented as a TPS, each Jacobian element is itself a TPS rather than a scalar quantity. The symplectic condition can therefore be evaluated order-by-order across the nonlinear map rather than just first order.

Symplecticity is tested using

$$J^T S J = S \,,$$

where

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} .$$

Since the Jacobian elements are TPS objects, the residual matrix

$$R = J^T S J - S$$

is also a TPS. Individual TPS coefficients of R directly quantify symplectic violation at each order. This allows for verification of symplecticity to arbitrary-order limited only by the chosen TPS truncation order.

WEAK-STRONG BEAM-BEAM SIMULATION

The TPS verification framework was applied to a weak-strong beam-beam simulation including longitudinal slicing of the strong beam. The strong beam is divided into equal-charge slices distributed longitudinally on the beam centroid. Weak beam particles receive sequential beam-beam kicks computed using the Bassetti-Erskine formulas [5] from each slice with drifts between collision points.

This model incorporates the hourglass effect through variation of the beta function across the interaction region, and the slingshot effect, both of which introduce an energy kick in the weak particles [3].

These effects are physically important for realistic beam-beam simulations but complicate analytic derivation of the full six-dimensional map. The slicing introduces multiple collision points whose combined symplecticity is difficult to manually verify, but can be easily verified through use of TPSA.

The TPS-based symplectic verification procedure can be implemented directly in Julia using a polymorphic tracking library such as SciBmad and a TPSA package such as `GTPSA.jl`. The following example in Listing 1 demonstrates the basic workflow. Initial phase-space coordinates are constructed as TPS variables, tracked through the beam-beam simulation, and used to construct the Jacobian matrix. The symplectic residual matrix is then computed directly from the TPS Jacobian.

Listing 1: TPS-based symplectic verification workflow in Julia. Uses the SciBmad structure for tracking and `GTPSA.jl` for TPSA implementation.

```julia
using GTPSA
using LinearAlgebra
using BeamTracking

# TPSA setup
order = 3
num_var = 6

d = Descriptor(num_var, order)

# Define expansion point
x0 = [0 0 0 0 0 0]

# Create bunch with TPS particle
b = Bunch(x0 + @vars(d)')

# Set beambeam params
params = (E_strong=1.0e11, charge=1.0,
    sig_x=0.2, sig_y=0.1, sig_z=1.0,
    N_particles=1.0e14, slices=3)

# Track through beam-beam element
BeamTracking.launch!(b.coords,
    KernelCall(BeamTracking.track_beambeam!,params))

# Final coordinate vector
xf = b.coords.v[1, :]

# Extract TPS Jacobian
J = Matrix{TPS}(undef, 6, 6)

for i in 1:6
    for j in 1:6
        J[i, j] = GTPSA.deriv(xf[i], j)
    end
end

# Canonical symplectic matrix
S = [
     0  1  0  0  0  0;
    -1  0  0  0  0  0;
     0  0  0  1  0  0;
     0  0 -1  0  0  0;
     0  0  0  0  0  1;
     0  0  0  0 -1  0
]

# Symplectic residual
R = transpose(J) * S * J - S

# Frobenius norm of linear part
err = norm(map(x -> GTPSA.scalar(x), R))
```

Since each matrix element of `J` is itself a TPS object, the residual matrix `R` retains information from the nonlinear part of the map. Individual TPS coefficients of `R` can therefore be viewed directly to identify symplectic violations at arbitrary nonlinear order.

RESULTS

TPS-generated Jacobians were computed for the weak-strong beam-beam simulation across varying incoming particle coordinates and TPS truncation orders. The symplectic condition was evaluated using the residual matrix

$$R = J^T S J - S\ ,$$

with the Frobenius norm of the residual used as the metric for symplectic error,

$$||R||_F = \sqrt{\sum_{ij} |R_{ij}|^2}\ .$$

Unlike ideal symplectic maps, the beam-beam model exhibited measurable symplectic error at all tested orders. The residual remained nonzero even as TPS truncation order increased, indicating that the observed error originates from the implemented beam-beam map itself rather than truncation effects in the TPS representation.

To study the dependence of the violation on phase-space amplitude, the Frobenius norm of the residual matrix was evaluated as a function of incoming horizontal coordinate x seen in Fig. 1. The symplectic error increased with incoming transverse amplitude, demonstrating that the deviation from symplecticity becomes stronger for particles farther from the reference trajectory.

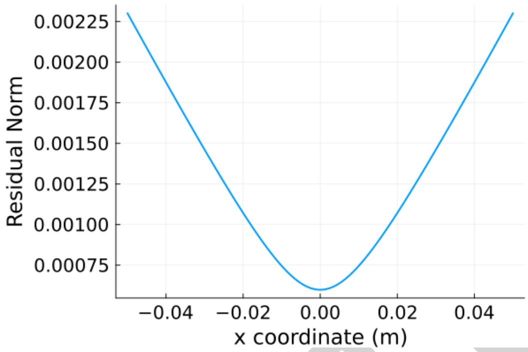

Figure 1: Frobenius norm of the symplectic residual matrix as a function of incoming horizontal coordinate x. The symplectic error increases with transverse amplitude, indicating a deviation from symplecticity in the beam-beam map.

The TPS framework additionally allowed the residual TPS coefficients to be inspected order-by-order. Nonzero coefficients were observed across multiple nonlinear orders, confirming that the violation is not restricted to the linearized map, but persists in the nonlinear part of the implemented beam-beam interaction.

These results demonstrate that TPS propagation can identify and accurately quantify nonlinear symplectic violations directly through tracking, even for complicated models whose analytic form is difficult to derive explicitly.

CONCLUSION

A TPS-based symplectic verification framework for accelerator tracking codes has been demonstrated. By propagating truncated power series variables directly through a polymorphic tracking library, Jacobians are computed automatically from the implemented physics model with no need for finite differencing.

The method was demonstrated on a weak-strong beam-beam simulation including equal-charge longitudinal slicing, hourglass effects, and slingshot effects. These features complicate analytic verification of symplecticity in the full six-dimensional map. TPS propagation instead verifies symplecticity directly through the tracking implementation.

The presented framework is broadly applicable to any polymorphic accelerator tracking software and provides a practical method for validating complex beam dynamics simulations.

REFERENCES

[1] Finite difference method, https://en.wikipedia.org/wiki/Finite_difference_method

[2] M. Signorelli, J. Devlin, G. H. Hoffstaetter, and D. Sagan, "SciBmad: A differentiable, GPU-parallelized software library for particle accelerator design, nonlinear analysis, and machine learning", presented at the IPAC'26, Deauville, France, May 2026, paper THP5325, this conference.

[3] D. Sagan, *The Bmad reference manual*, 2026. https://www.classe.cornell.edu/bmad/manual.html

[4] M. Signorelli, L. Deniau, G. Hoffstaetter, D. Sagan, and O. Beznosov, "GTPSA.jl: A SciBmad interface to the generalised truncated power series algebra library", in *Proc. NAPAC'25*, Albuquerque, NM, USA, Aug. 2025, pp. 145-148. doi:10.18429/JACoW-NAPAC2025-MOP044

[5] M. Bassetti and G. A. Erskine, "Closed expression for the electrical field of a two-dimensional Gaussian charge", CERN, Geneva, Switzerland, Rep. CERN-ISR-TH-80-06, 1980. https://cds.cern.ch/record/122227