

|| OUTLINE ||

|| Part I – Classical computational methods to solve the Einstein's equations ||

- ⚙ The physical problem
- ⚙ How does one solve Einstein's equations?
- ⚙ Solutions to perturbative approaches via traditional computational methods
- ⚙ Challenges with SOTA methods in GW astronomy

|| Part II – Introducing Geometric Numerical Machine Learning and SINFONIA ||

- ⚙ Introduction to key concepts in AI and ML
- ⚙ Can ML aid address current challenges faced by SOTA trad methods :: the case for Gravity-Informed Neural Network (GINNs)?
- ⚙ How to craft a GINNs? Cruising through different GINNs via educated proxy model
- ⚙ Introducing (FINALLY):
 - ⚙ SINFONIA-J0
 - ⚙ SINFONIA-J1for modelling of BBH binaries.

|| THE PHYSICAL PROBLEM :::: THE 2*BODY PROBLEM VIA EINSTEIN'S EQUATIONS ||

My background in a nutshell: Computational relativity.

My focus essentially is solving **Einstein's evolution** equations:

$$G_{\alpha\beta} + \Lambda g_{\alpha\beta} = \kappa T_{\alpha\beta}$$

We can do this in various ways:

🔧 Perturbative methods:

🔧 PN/PM Expansions

🔧 BHPT/GSF → My PhD

🔧 Numerical relativity

🔧 Also, clever hybrid methods such as **EOB**

🔧 Machine Learning and Artificial Intelligence

L.J. GOMES DA SILVA

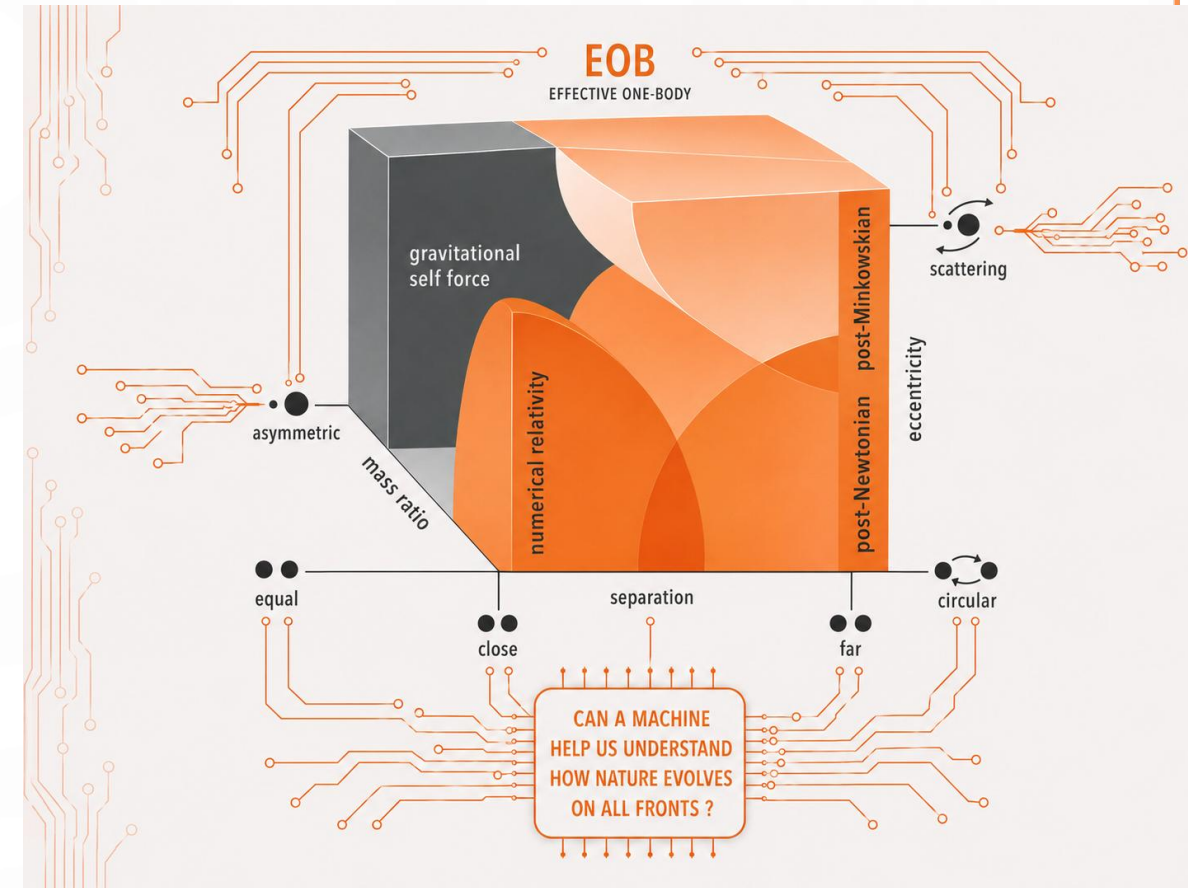


Figure 1 - Schematic range of validity of the methods to solve the relativistic two-body problem. Edited from **Ana Carvalho's** work for <https://gwsky.org/research/>.

31/05/2026

3

* & more

|| MY RESEARCH BACKGROUND IN A NUTSHELL ||

🔧 Development of numerical methods for solving the Einstein equations and beyond :

During my PhD:

$$[-\partial_t^2 + \partial_x^2 - V^{S/RW/Z}(r)]\phi^{S/RW/Z}(t, r) = S^{S/RW/Z}(t, r) = G(t, r)\delta(r - r_p(t)) + I(t, r)\delta'(r - r_p(t))$$

🔧 Development of numerical methods, **DiscoTEX** and **ANSORG** to solve the governing equations of Extreme-Mass-Ratio-Inspirals:

- 🔧 Hyperboloidal methods
- 🔧 Discontinuous collocation methods
- 🔧 Multi-domain spectral spatial discretisation methods
- 🔧 Discontinuous time-integration methods:
 - ⚙️ Implicit and implicit-turned-explicit
 - ⚙️ Multi-domain spectral time-integration (**ANSORG**)

🔧 Multi-domain spectral spatial discretisation methods

During my post-doc:

- 🔧 Geometric Numerical Machine Learning
- 🔧 Development of numerical methods for evolving pathological beyond GR equations such as those of k-essence.
- 🔧 Self-consistent evolutions for the modelling of EMRIs

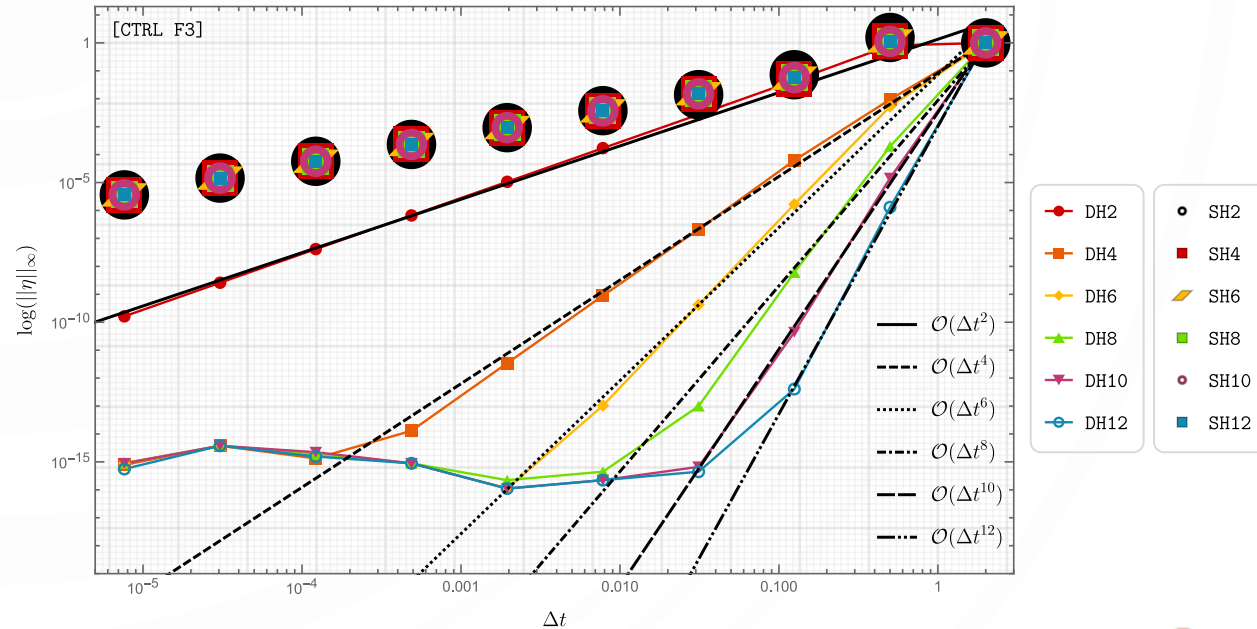


Figure 2 – DiscoTEX, see: <https://arxiv.org/pdf/2401.08758>, <https://arxiv.org/pdf/2411.14399>.

|| POST NEWTONIAN APPROXIMATION ||

⚙️ **Physical regime of applicability:** Weak field, inspiral, before merger

$$\epsilon \sim \frac{v^2}{c^2} \sim \frac{GM}{rc^2} \ll 1$$

⚙️ **Start from Einstein's equations**

$$G_{\alpha\beta} + \Lambda g_{\alpha\beta} = \kappa T_{\alpha\beta}$$

⚙️ **Expand around the Newtonian limit**

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad h_{\mu\nu} = \mathcal{O}\left(\frac{v^2}{c^2}\right)$$

⚙️ **The dynamics becomes an expansion in powers of $1/c$**

$$H = Mc^2 + H_N + \frac{1}{c^2} H_{1PN} + \frac{1}{c^3} H_{1.5PN} + \frac{1}{c^4} H_{2PN} + \frac{1}{c^5} H_{2.5PN} + \frac{1}{c^6} H_{3PN} + \frac{1}{c^7} H_{3.5PN} + \dots$$

PN order	Scaling	Enters through	Main physical effect
0PN	c^0	H_N	Newtonian two-body motion
1PN	c^{-2}	H_{1PN}	Conservative relativistic correction; periastron advance
1.5PN	c^{-3}	H_{SO}	Leading spin-orbit coupling; spin precession
2PN	c^{-4}	H_{2PN}, H_{SS}	Higher conservative orbital terms; leading spin-spin effects
2.5PN	c^{-5}	$\mathbf{F}_{2.5PN}$	Leading radiation reaction; inspiral begins

|| MODELLING SPINNING BBH BINARIES VIA PN AND GEOMETRICAL NUMERICAL INTEGRATION ||

⚙️ We are interested in modelling spinning BBH binaries accurately

Dynamics can be accurately captured by PN equations of motion:

$$\dot{\mathbf{X}} = \frac{\partial H}{\partial \mathbf{P}} \quad \dot{\mathbf{P}} = -\frac{\partial H}{\partial \mathbf{X}} + \mathbf{F} \quad \dot{\mathbf{S}}_a = \frac{\partial H}{\partial \mathbf{S}_a} \times \mathbf{S}_a$$

Then:

$$H_{\text{ADM}}(\mathbf{X}, \mathbf{P}, \mathbf{S}_1, \mathbf{S}_2) = H_{\text{ADM}}^{\text{orb}}(\mathbf{X}, \mathbf{P}) + H_{\text{ADM}}^{\text{spin}}(\mathbf{X}, \mathbf{P}, \mathbf{S}_1, \mathbf{S}_2).$$

And:

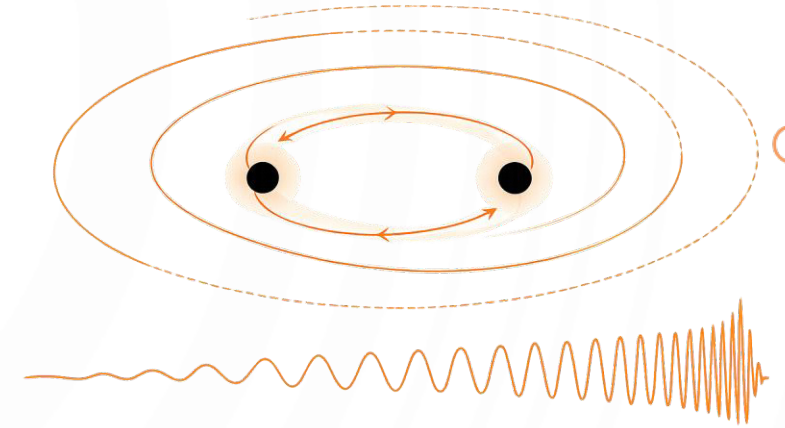
$$H_{\text{Orb}} = H_N + H_{1\text{PN}} + H_{2\text{PN}} + H_{3\text{PN}}$$

$$H_{\text{Spin}} = H_{\text{SO}} + H_{S_1 S_1} + H_{S_1 S_2} + H_{S_2 S_2}$$

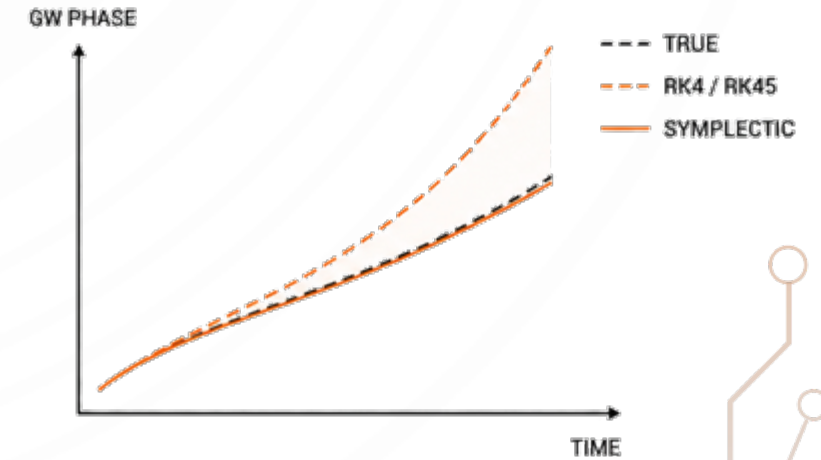
Challenges:

- ⚙️ Extremely long-integrations, 10^5 to 10^6
- ⚙️ Inherent multi-scale structure
- ⚙️ Need to scan over wide regions of parameter space
- ⚙️ Chaos
- ⚙️ Waveform accuracy and fidelity

INSPIRALLING BINARY (PN DYNAMICS)



PHASE DRIFT OVER LONG INTEGRATIONS



➔ The spin terms enter through spin-orbit and spin-spin couplings:

$$\text{Spin-Orbit (SO): } H_{SO} = \frac{2G}{c^2} \frac{S_{\text{eff}} \cdot L}{R^3} \quad \text{Spin-Spin (SS): } H_{S_1 S_2} = \frac{G}{c^2 R^3} [3(S_1 \cdot N)(S_2 \cdot N) - S_1 \cdot S_2]$$

With orbital angular
momentum:

$$L = X \times P$$

And the effective spin
contribution

$$S_{\text{eff}} = \left(1 + \frac{3m_2}{4m_1}\right) S_1 + \left(1 + \frac{3m_1}{4m_2}\right) S_2$$

➔ This yields not just any arbitrary set of ODEs but it has Poisson structure. For orbital variables we have the usual canonical bracket.

➔ while for spins we have a Lie-Poisson bracket:

$$\{A, B\} = \frac{\partial A}{\partial X} \cdot \frac{\partial B}{\partial P} - \frac{\partial A}{\partial P} \cdot \frac{\partial B}{\partial X} + \sum_{a=1}^2 \det\left(\frac{\partial A}{\partial S_a}, S_a, \frac{\partial B}{\partial S_a}\right) \quad \text{where: } A = (\mathbf{X}, \mathbf{P}, \mathbf{S}_1, \mathbf{S}_2), \text{ and } B = H_i$$

Thus:

- 🔧 Structure encodes both Hamiltonian orbital motion and spin precession;
- 🔧 It implies magnitudes should remain constant and the conservative dynamics should preserve the appropriate phase space geometry.

🔧 Hamiltonian splitting:

$$H = H_{\text{Orb}} + H_{\text{SO}} + H_{\text{SS}}$$

🔧 Split the system into sub-flows and compose:

$$\Phi_h \approx \Phi_{h/2}^{\text{SS}*} \circ \Phi_{h/2}^{\text{SO}*} \circ \Phi_h^{\text{Orb}} \circ \Phi_{h/2}^{\text{SO}} \circ \Phi_{h/2}^{\text{SS}}$$

where:

$$\Phi_h^{\text{Orb}} = \Phi_{h/2}^{\text{PN},*} \circ \Phi_h^{\text{N}} \circ \Phi_{h/2}^{\text{PN}}$$

🔧 Spin subflows reduce to exact rotations. Spin equation has the form: $\dot{S} = \Omega \times S$

▶ whose solution is a rigid body rotation: $S(t) = R(\Omega, t) S(0)$

▶ Thus, the SO and SS parts are decomposed into analytically solvable subflows:

$$H_{\text{SO}} = H_{\text{SO}}^1 + H_{\text{SO}}^2 + H_{\text{SO}}^3$$

$$H_{\text{SO}}^i = S_{\text{eff}}^i L^i / R^3$$

▶ It preserves $|S|^2$ exactly, up to numerical precision.

$$\Phi_{h/2}^{H_{\text{SO}}} = \varphi_{h/2}^{H_{\text{SO}}^3} \circ \varphi_{h/2}^{H_{\text{SO}}^2} \circ \varphi_{h/2}^{H_{\text{SO}}^1}$$

$$\Phi_{h/2}^{H_{\text{SO}},*} = \varphi_{h/2}^{H_{\text{SO}}^1} \circ \varphi_{h/2}^{H_{\text{SO}}^2} \circ \varphi_{h/2}^{H_{\text{SO}}^3}$$

$$\Phi_{h/2}^{H_{\text{SS}}} = \varphi_{h/2}^D \circ \varphi_{h/2}^C \circ \varphi_{h/2}^B \circ \varphi_{h/2}^A$$

🔧 Graduated to higher-orders via further compositions:

$$\Phi_h^{(4)} = \Phi_{\gamma_5 h} \circ \Phi_{\gamma_4 h} \circ \Phi_{\gamma_3 h} \circ \Phi_{\gamma_2 h} \circ \Phi_{\gamma_1 h}$$

$$\gamma_1 = \gamma_2 = \gamma_4 = \gamma_5 = \frac{1}{4 - 4^{1/3}}$$

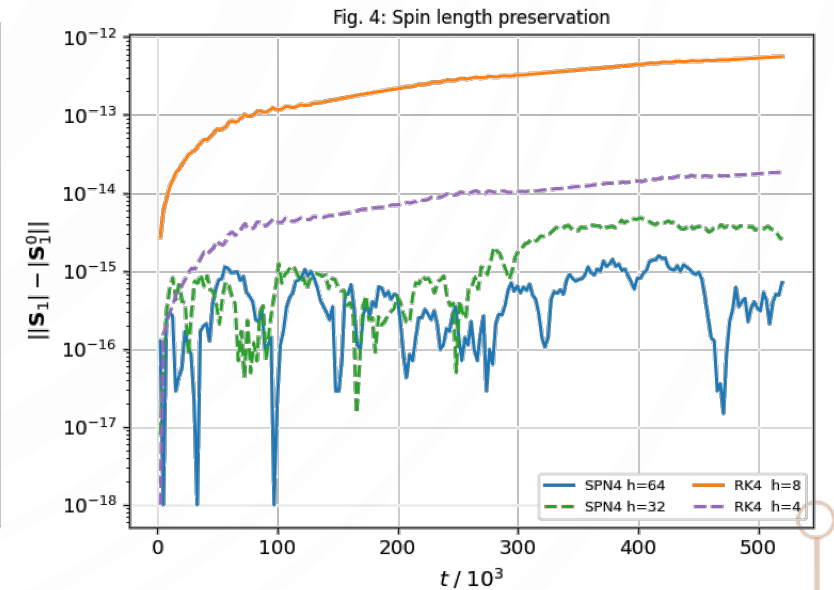
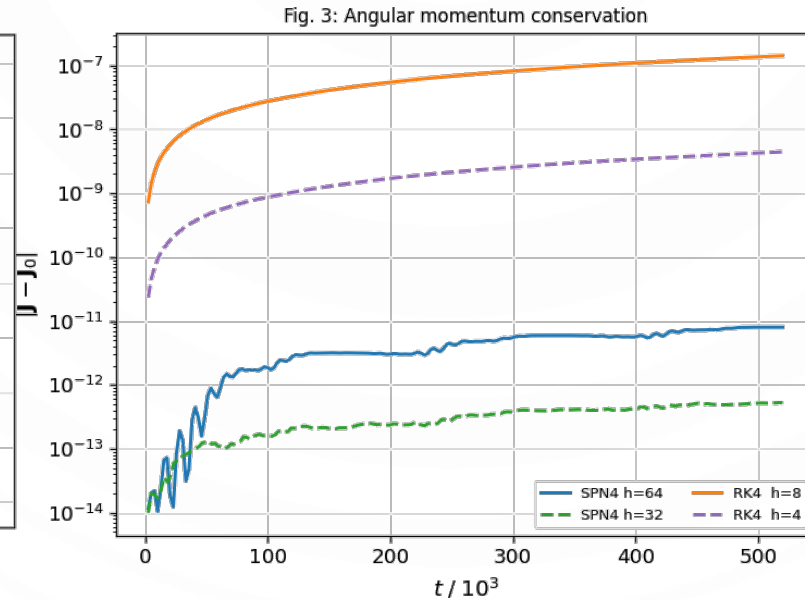
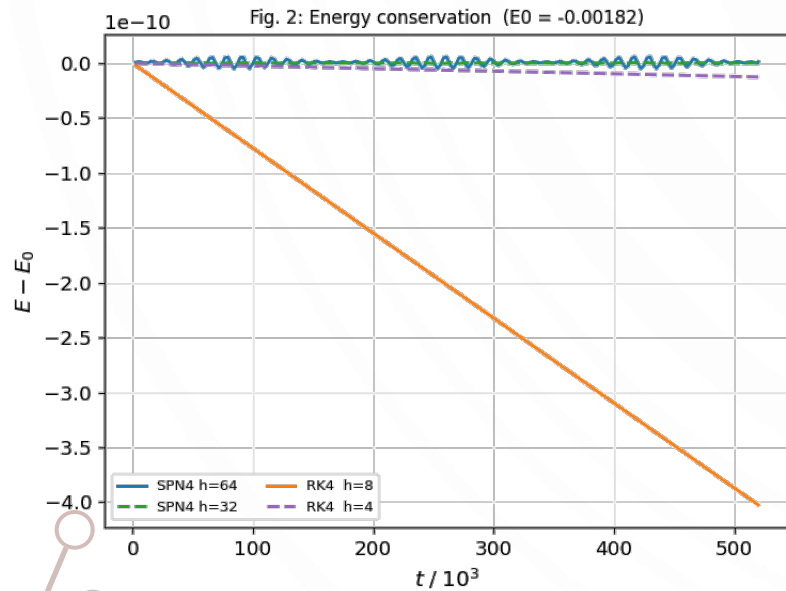
$$\gamma_3 = -\frac{4^{1/3}}{4 - 4^{1/3}}$$

|| GEOMETRIC TIME INTEGRATION OF CONSERVATIVE DYNAMICS || RESULTS ||

➔ Features of the geometric Poisson integrator:

- ⚙ Structure preserving, preserves symplectic structure
- ⚙ Demonstrates long-term stability with no phase-drift unlike explicit phase drifting schemes
- ⚙ Allows for larger time-stepping

➔ Results:



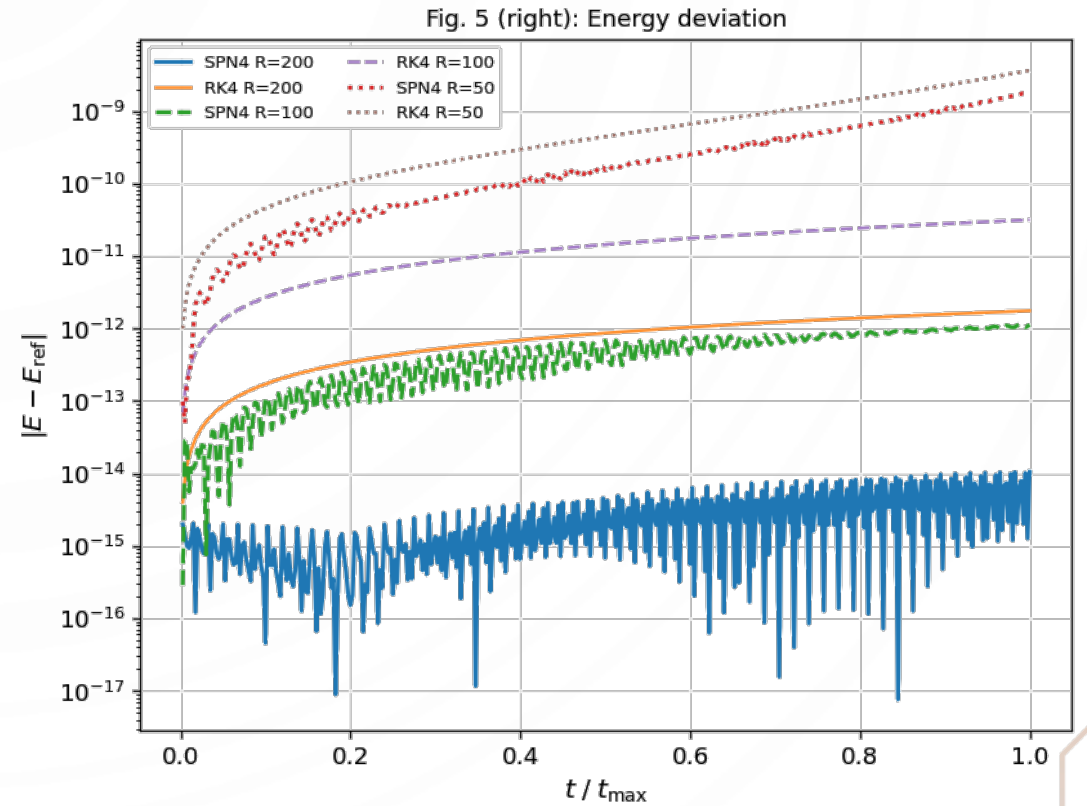
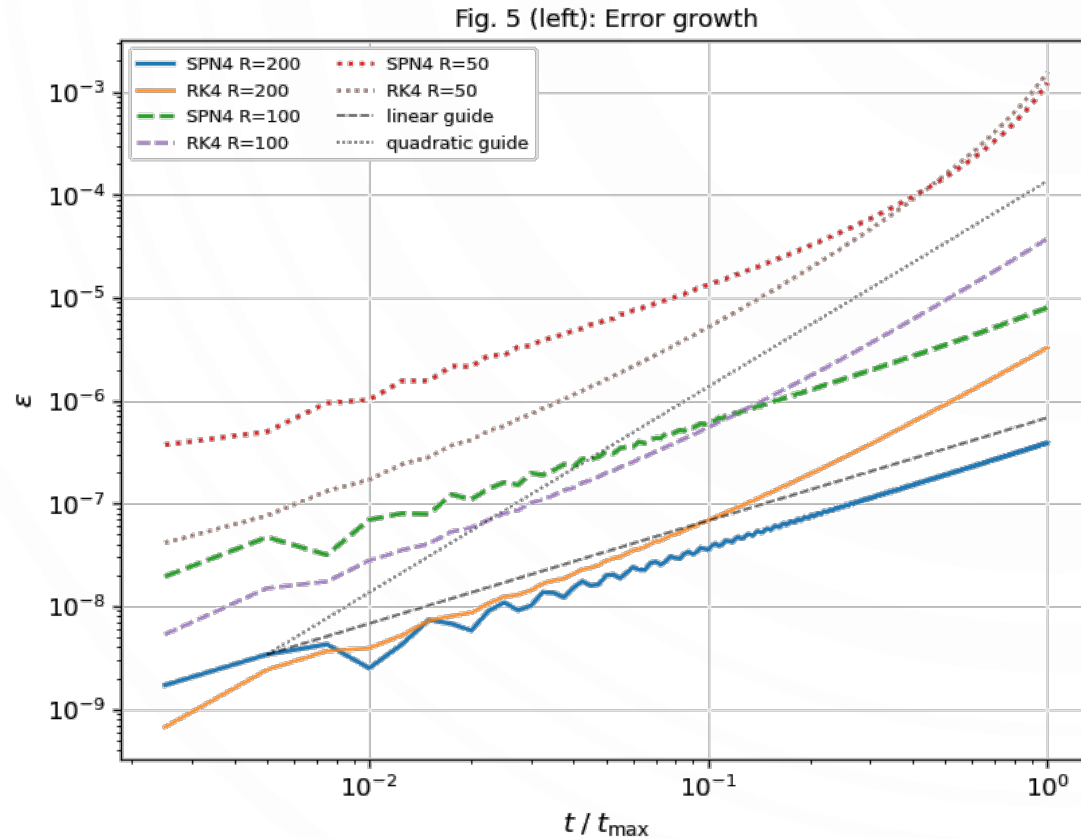
Reproduced from <https://arxiv.org/pdf/1003.5122> in JAX.

|| GEOMETRIC TIME INTEGRATION || DISSIPATIVE DYNAMICS || RESULTS ||

⚙ Dissipative contribution padded around the conservative flow: $\Phi_h^{\text{diss}} = \Phi_{h/2}^F \circ \Phi_h^{\text{cons}} \circ \Phi_{h/2}^F$

⚙ Modelled by a Euler Step: $P' = P + \frac{h}{2} F(X, P, \dot{X})$

⚙ Results:



L.J. GOMES DA SILVA

Reproduced from <https://arxiv.org/pdf/1003.5122> in JAX.

➡ Our scheme in a nutshell: $H_{\text{ADM}} = H_{\text{orb}} + H_{\text{spin}}$ with radiation reaction added as $\dot{\mathbf{P}} = -\frac{\partial H_{\text{ADM}}}{\partial \mathbf{X}} + \mathbf{F}_{\text{RR}}$.

➡ Features of previous dissipative geometric scheme:

- 🔧 Respects geometry of the PN system;
- 🔧 Long-term stability and reduced phase-drift;
- 🔧 Inspiral produced by adding a dissipative momentum update;
- 🔧 Breaks symplectic structure preservation.

➡ Can we treat the dissipative sector more structurally ???

➡ Asked and answered by Galley and collaborators in: <https://arxiv.org/abs/1210.2745>, <https://arxiv.org/abs/1412.3082> and <https://arxiv.org/abs/1506.08443>.

🔧 Solution:

- ▶ Instead of writing radiation reaction only as an external force $\mathbf{F}$, we introduce a non-conservative potential K in an augmented/doubled phase space, $\mathbf{F}_{\text{RR}} \rightarrow K$.
- ▶ Or more explicitly,

$$\mathbf{F}_{\text{RR}} = \left[\frac{\partial K}{\partial \mathbf{q}_-} - \frac{d}{dt} \frac{\partial K}{\partial \dot{\mathbf{q}}_-} \right]_{\text{PL}},$$

The dissipative force then comes from K , in the same way that the conservative force comes from H !

⚙️ Radiation reaction at 2.5PN: a non-conservative force **not derivable** from any single Lagrangian!

💡 **Introduce Galley's trick: Double variables for non-conservative dynamics.** 💡

⚙️ Galley's idea is to extend the variational principle to **two histories**: (q_+, q_-)

and define the action

$$S[q_+, q_-] = \int dt [L(q_+, \dot{q}_+) - L(q_-, \dot{q}_-)] + \int dt K(q_+, q_-, \dot{q}_+, \dot{q}_-).$$

Here,

- ▶ $L(q_+, \dot{q}_+) - L(q_-, \dot{q}_-)$ describes the doubled conservative system,
- ▶ while $K(q_+, q_-, \dot{q}_+, \dot{q}_-)$ encodes the non-conservative interaction.

⚙️ Taking the variation and then applying the **physical limit**: $q_- \rightarrow 0, \dot{q}_- \rightarrow 0$,

⚙️ Produces the ordinary Euler–Lagrange equation plus a dissipative force:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \left. \frac{\partial K}{\partial q_-} \right|_{q_-=0, \dot{q}_-=0}.$$

⚙️ So, the non-conservative force is generated by

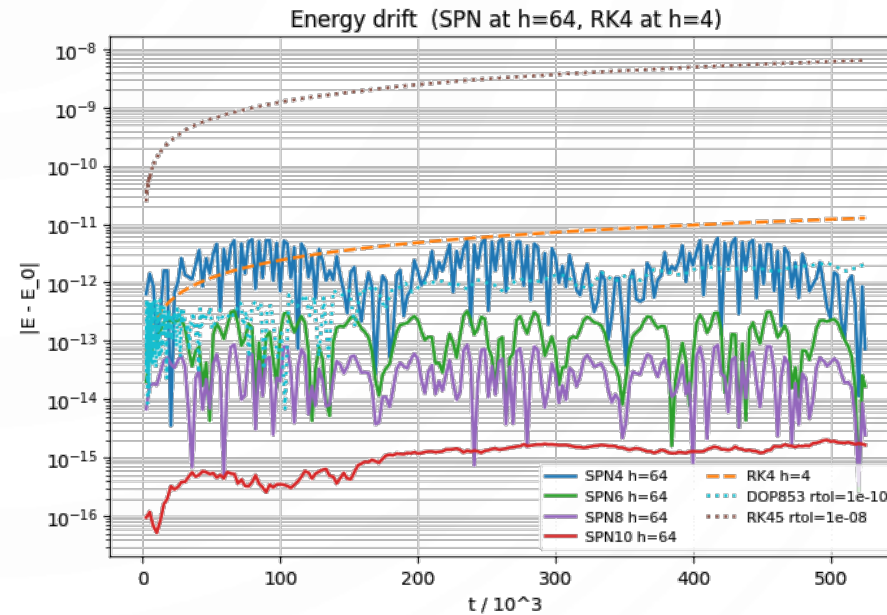
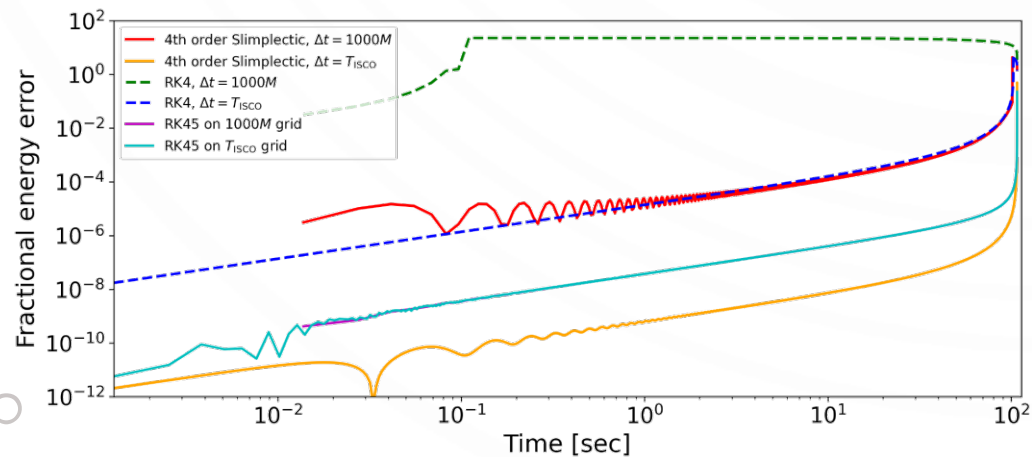
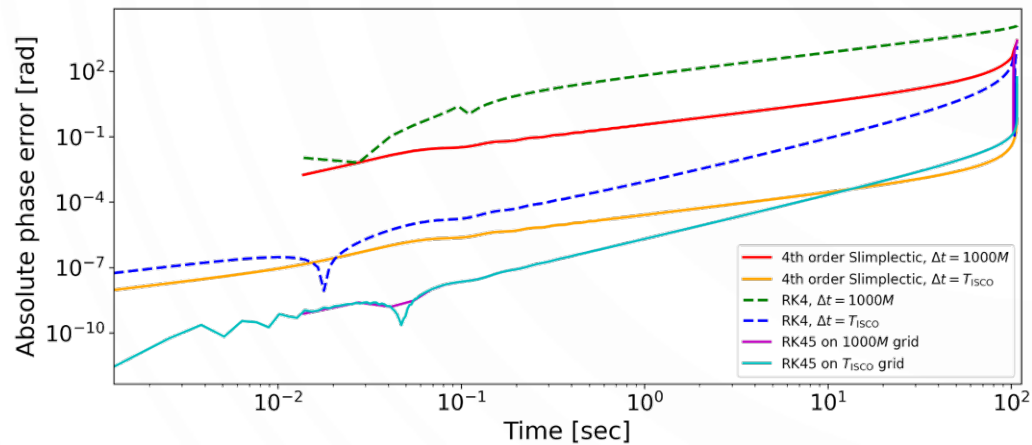
$$F_K = \left. \frac{\partial K}{\partial q_-} \right|_{\text{physical limit}}.$$

From: <https://arxiv.org/pdf/1506.08443>

|| GEOMETRIC TIME INTEGRATION || CONCLUSIONS ||

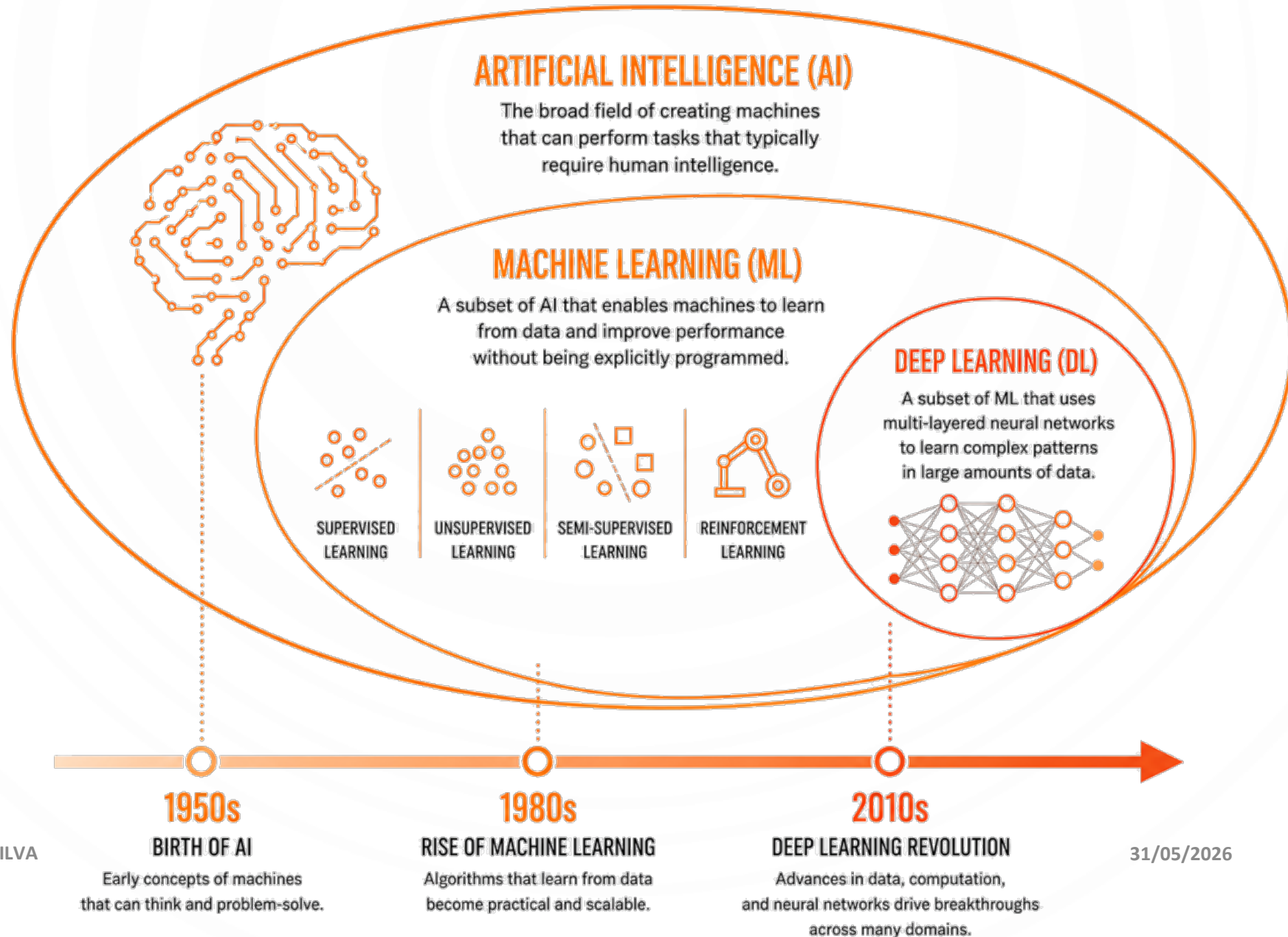
» Features of symplectic and symplectic schemes:

- 🔧 Controlled breaking of structure-preservation
- 🔧 Long-term stability
- 🔧 Accurate results without the need to resort to adaptive methods!

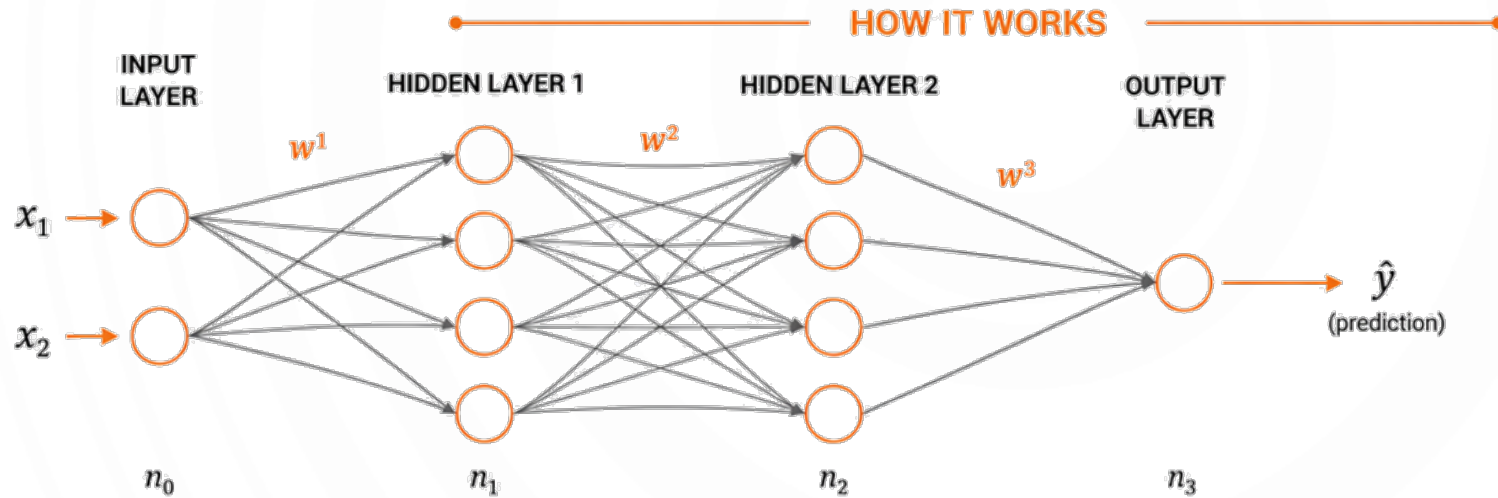


Reproduced from <https://arxiv.org/abs/1506.08443> (left), <https://arxiv.org/pdf/1003.5122> (right), in JAX

MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

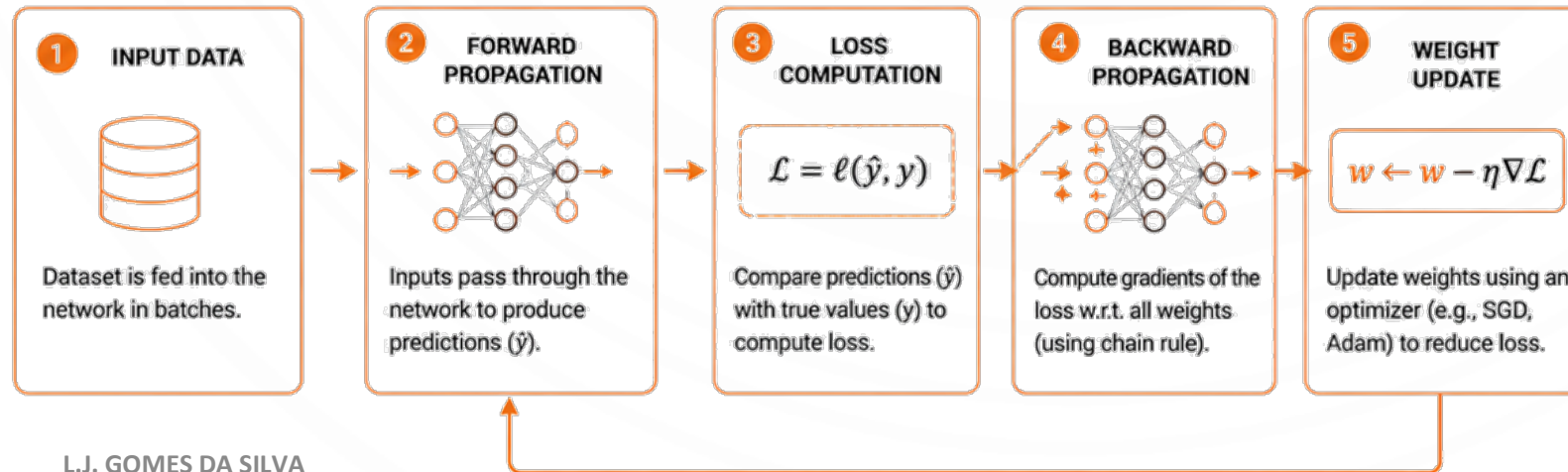


ML AND AI || WHAT IS A NEURAL NETWORK ?



- Neurons (nodes)**
Process information
- Connections (weights)**
Learnable parameters
- Activation Function**
Introduces non-linearity
- n_i **Number of neurons in layer i**

THE TRAINING PROCESS (HOW NEURAL NETWORKS LEARN)



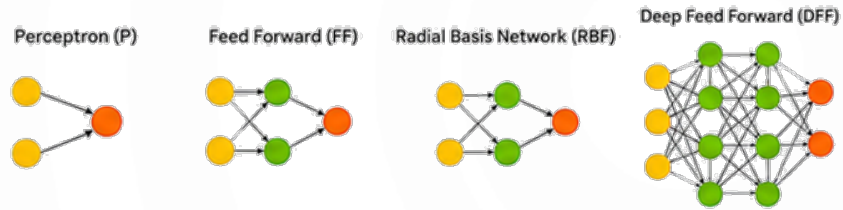
- KEY COMPONENTS**
- Activation Functions**
Introduce non-linearity (e.g., ReLU, tanh, σ).
 - Weights (W)**
Learned during training to minimize loss.
 - Biases (b)**
Allow flexibility in learning patterns.
 - Loss Function**
Quantifies error between prediction and truth.

ML AND AI || A KALEIDOSCOPE OF NN ARCHITECTURES ||

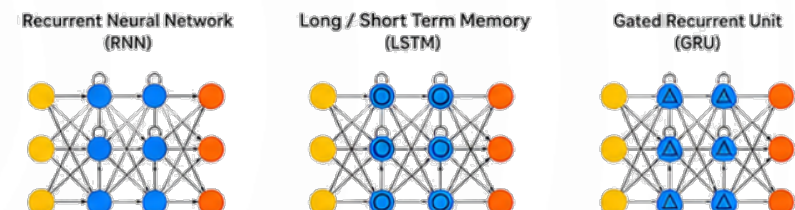
LEGEND

- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool

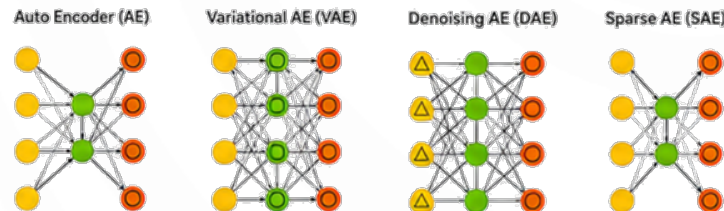
BASIC FEEDFORWARD NETWORKS



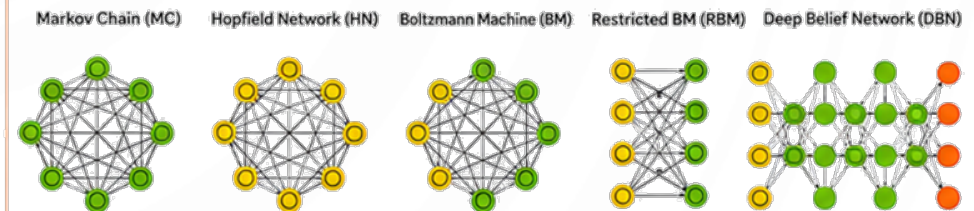
RECURRENT & MEMORY NETWORKS



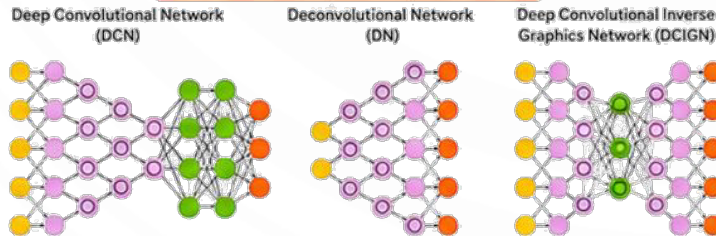
AUTOENCODERS



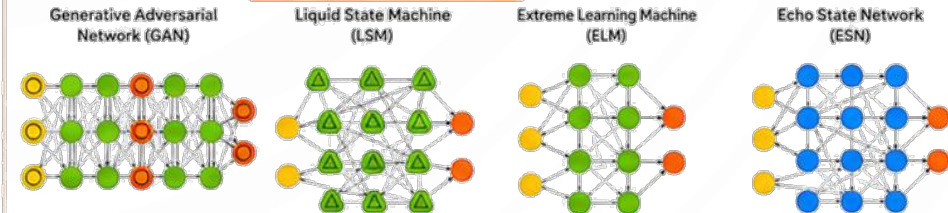
ENERGY-BASED & PROBABILISTIC MODELS



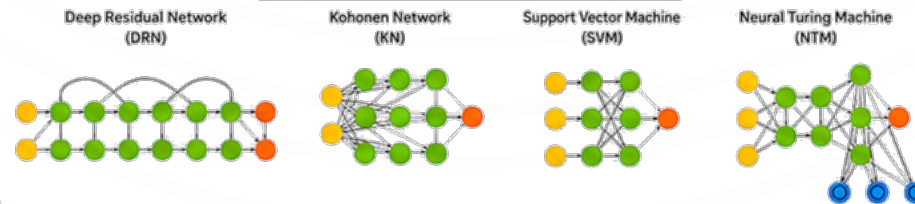
CONVOLUTIONAL ARCHITECTURES



GENERATIVE MODELS



OTHER IMPORTANT ARCHITECTURES



L.J. GOMES DA SILVA

31/05/2026

16

Inductive bias ▶ built-in assumptions that guide learning.

A gravity-informed neural network (GINN) is a neural model whose architecture, loss, or outputs are constrained by known gravitational structure.

Examples:

- ⚙ Hamiltonian flow
- ⚙ Radiation reaction,
- ⚙ Symmetries, conservation
- ⚙ Conservation Laws
- ⚙ Phase coherence

GINN constrained by 2-body structure:

$$z = (X, P, S_1, S_2), \lambda = (m_1, m_2, \chi_1, \chi_2, \dots)$$

$$\dot{z} = f_{\text{PN/EOB}}(z; \lambda) + f_{\theta}(z; \lambda)$$

Example: orbital + dissipative structure

$$\dot{X} = \frac{\partial H_{\text{PN/EOB}}}{\partial P},$$

$$\dot{P} = -\frac{\partial H_{\text{PN/EOB}}}{\partial X} + F_{\text{RR}} + F_{\theta}.$$

$$\dot{S}_a = \Omega_a(X, P, S_1, S_2) \times S_a + G_{\theta, a}.$$



For next-gen GW modelling we need:

- ⚙ Long-time phase accuracy over millions of orbits;
- ⚙ Fast parameter sweeps;
- ⚙ Out-of-distribution capabilities (we do not want to just fit training waveforms. We want to design an extrapolator);
- ⚙ Long-time stability.

🔧 **Consider a proxy toy model:** Schwarzschild Binet equation: $u'' + u = \frac{M}{L^2} + 3Mu^2$

$$u(\phi) = \frac{1}{r(\phi)}$$

► Introducing $v(\phi) = \frac{du}{d\phi}$

s.t.

$$u' = v,$$

$$v' = \frac{M}{L^2} + 3Mu^2 - u$$

Good proxy because:

⚙️ because captures a key gravitational effect: relativistic periastron precession!!

⚙️ **Conserved Invariant:**

$$I(u, v) = \frac{1}{2}v^2 + \frac{1}{2}u^2 - \frac{M}{L^2}u - Mu^3$$

$$\frac{dI}{d\phi} = 0$$

⚙️ **Precession diagnostic:** $\Delta\phi_{\text{rad}} = 2\pi + \delta, \quad \delta > 0$

⚙️ **The orbit evolves on the invariant curve:**

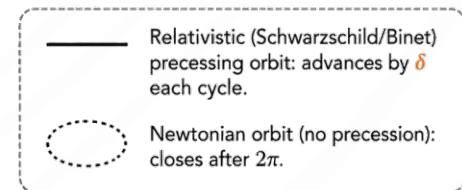
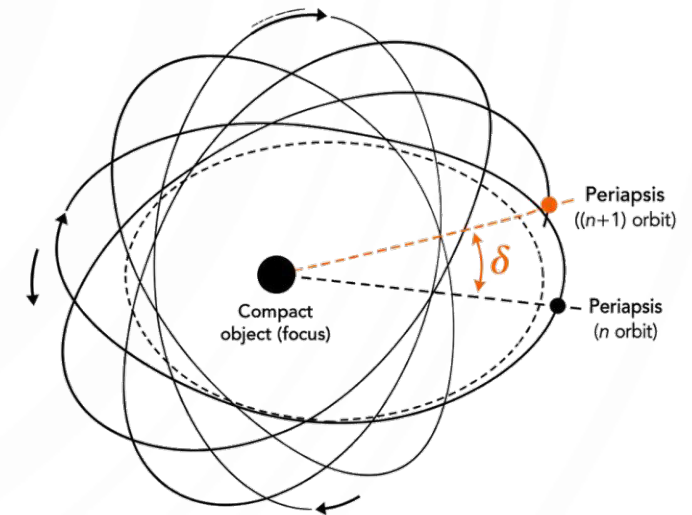
$$I(u, v) = I_0$$

Small invariant error shall give:

$$I(u, v) \neq I_0$$

$$\Delta\phi = 2\pi + \delta$$

δ = periastron advance per orbit



🔧 Equation bias

Enforce the local equations of motion:

$$\begin{aligned} r_1(\phi) &= u'_\theta(\phi) - v_\theta(\phi) \\ r_2(\phi) &= v'_\theta(\phi) - \left(\frac{M}{L^2} + 3Mu_\theta(\phi)^2 - u_\theta(\phi) \right) \\ \mathcal{L}_{\text{phys}} &= \frac{1}{N} \sum_{i=1}^N [r_1(\phi_i)^2 + r_2(\phi_i)^2] \end{aligned}$$

🔧 Invariant bias

Preserve the geometric manifold:

$$I(u_{\theta'}(\phi), v_\theta(\phi)) = I_0$$

or softly,

$$\mathcal{L}_I = \frac{1}{N} \sum_{i=1}^N [I(u_{\theta'}(\phi_i), v_\theta(\phi_i)) - I_0]^2.$$

🔧 Flow bias

Learn the evolution map itself:

$$\begin{aligned} z &= (u, v) \\ z_{n+1} &= \Phi_h^\theta(z_n) \end{aligned}$$

For symplectic flow maps,

$$(\partial_z \Phi_h^\theta)^T J (\partial_z \Phi_h^\theta) = J.$$

Method class	What is learned?	Proxy-model example	Known NN architectures
Vector-field learning	The differential law $\dot{z} = f_\theta(z)$	Learn $u' = v$, $v' = F_\theta(u, v)$ or $H_\theta(u, v)$	PINNs, Neural ODEs , HNNs
Operator learning	A map from problem data to solution functions	$(u_0, v_0) \mapsto (u(\phi), v(\phi))$	DeepONet, Fourier Neural Operator
Flow-map learning	The time-step/evolution map directly	$z_n \mapsto z_{n+1}$, or $(z_0, \phi) \mapsto z(\phi)$	SympFlow, SympNets , neural integrators

⚙️ Learn the local rule of motion:

$$\frac{dz}{d\phi} = f(z)$$

|| VFL via traditional PINN ||

The network directly outputs the trajectory:

$$\phi \mapsto (u_{\theta}'(\phi), v_{\theta}(\phi))$$

and is trained by residual minimization:

$$\begin{aligned} r_1 &= u_{\theta}' - v_{\theta} \\ r_2 &= v_{\theta}' - \left(\frac{M}{L^2} + 3Mu_{\theta}^2 - u_{\theta} \right). \\ \mathcal{L}_{\text{PINN}} &= \frac{1}{N} \sum_{i=1}^N [r_1(\phi_i)^2 + r_2(\phi_i)^2]. \end{aligned}$$

or approximate it by

$$\frac{dz}{d\phi} = f_{\theta}(z).$$

|| VFL via HNN ||

Instead of learning the vector field directly, learn a scalar Hamiltonian-like object:

$$H_{\theta}(z)$$

and obtain the vector field by differentiation:

$$\dot{z} = J \nabla H_{\theta}(z).$$

For this reduced conservative system, the invariant plays the role of the Hamiltonian:

$$H(z) \sim I(w, v).$$

So, one can write

$$\dot{z} = J \nabla I(z).$$

🔧 For the Schwarzschild Binet system,

$$z = \begin{pmatrix} u \\ v \end{pmatrix}, \quad f(z) = \begin{pmatrix} v \\ \frac{M}{L^2} + 3Mu^2 - u \end{pmatrix}.$$

|| References ||

📄 PINNs:

<https://arxiv.org/abs/1711.10561>,
<https://arxiv.org/abs/1711.10566>

📄 HNNs:

<https://arxiv.org/abs/1906.01563>

- Do not learn one trajectory, we learn the solution operator: $\mathcal{G}: (u_0, v_0) \mapsto (u(\phi), v(\phi))$.
So instead of training a new model for each orbit, one trained model can represent a family of orbits.

DeepONet structure

DeepONet splits the problem into two parts:

Branch network

The branch network receives the initial condition: (u_0, v_0)

and produces coefficients:

$$b_k(u_0, v_0), k = 1, \dots, p.$$

Trunk network

The trunk network receives the coordinate: ϕ

and produces learned basis functions:

$$T_k(\phi), k = 1, \dots, p.$$

Operator reconstruction

For two outputs, u and v ,

$$u_\theta(\phi; u_0; v_0) = u_0 + s(\phi) \sum_{k=1}^p b_k(u_0, v_0) T_{u,k}(\phi)$$

$$v_\theta(\phi; u_0; v_0) = v_0 + s(\phi) \sum_{k=1}^p b_k(u_0, v_0) T_{v,k}(\phi)$$

with the hard initial-condition factor

$$s(\phi) = \frac{\phi - \phi_0}{\phi_f - \phi_0}.$$

Therefore,

$$s(\phi_0) = 0$$

and so

$$u_\theta(\phi_0) = u_0, v_\theta(\phi_0) = v_0.$$

Loss function:

Even though the architecture is operator-based, we can still impose the equation residuals:

$$\begin{aligned} r_1 &= u'_\theta - v_\theta \\ r_2 &= v'_\theta - \left(\frac{M}{L^2} + 3Mu_\theta^2 - u_\theta \right) \end{aligned}$$

$$\mathcal{L}_{\text{DeepONet}}$$

$$= \frac{1}{N} \sum_{i=1}^N [r_1(\phi_i)^2 + r_2(\phi_i)^2].$$

Instead of learning the derivative, we learn the time-advance maps: $\dot{z} = f(z)$.

This induces: $z(\phi + h) = \Phi_h(z(\phi))$.

So, the network learns: $\Phi_h^\theta: z_n \mapsto z_{n+1}$.

For the Binet proxy

$$z_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}, \quad z_{n+1} = \Phi_h^\theta(z_n).$$

The desired property is

$$I(z_{n+1}) \approx I(z_n).$$

Over many steps,

$$I(z_n) \approx I_0, n = 0, 1, \dots, N.$$

|| For Hamiltonian systems ||

A symplectic flow map satisfies

$$(\partial_z \Phi_h^\theta)^T J (\partial_z \Phi_h^\theta) = J.$$

Thus, map preserves the symplectic geometry of phase space!

A standard solver repeatedly evaluates, $\dot{z} = f(z)$ and integrates forward.

A learned flow map directly evaluates, $z_{n+1} = \Phi_h^\theta(z_n)$. So the learned object is closer to a numerical integrator than to a local force model!

Type of learning used in **SINFONIA**,
method, more details ahead.

- ▶ A vanilla PINN may satisfy $r_1 \approx 0, r_2 \approx 0$, but still drift in the invariant: $I(u_{\theta'}(\phi), v_{\theta}(\phi)) \neq I_0$.
- ▶ For long-time relativistic orbits, this matters because ::::: invariant drift $\Rightarrow$ wrong radial period $\Rightarrow$ wrong precession.

➡ Can we improve the inductive bias of the PINNs?

Attempt 1: Add an invariant penalty to the loss function! Aka soft-conservative PINNs

$$\mathcal{L}_{\text{soft-PINN}} = \mathcal{L}_{\text{phys}} + \lambda_I \mathcal{L}_I$$

where

$$\mathcal{L}_I = \frac{1}{N} \sum_{i=1}^N [I(u_{\theta'}(\phi_i), v_{\theta}(\phi_i)) - I_0]^2.$$

This encourages the network to remain close to $I(u, v) = I_0$.

L.J. GOMES DA SILVA

Attempt 2: Borrow from traditional geometrical time-integration methods and project the invariant back onto the manifold

Instead of only penalizing invariant error, project the prediction back to the invariant manifold.

Let

$$z_{\theta} = (u_{\theta'}, v_{\theta}).$$

The invariant manifold is

$$\mathcal{M}_{I_0} = \{z: I(z) = I_0\}.$$

The projected state is

$$z_c = z_{\theta} + \lambda \nabla I(z_{\theta}),$$

where λ is chosen so that

$$I(z_c) = I_0.$$

See <https://arxiv.org/abs/2311.14131>.

31/05/2026

23

- ▶ Combine operator learning with invariant preservation as with cPINNs.
- ▶ The operator is: $\mathcal{G}_\theta: (u_0, v_0) \mapsto z_\theta(\phi; u_0; v_0)$.
- ▶ The conservative correction is: $z_\theta \mapsto z_c = z_\theta + \lambda \nabla I(z_\theta)$ with $I(z_c) = I_0$.

Continuation strategy

- ▶ Early in training, projection can destabilize optimization. So, use a gradual projection blend:

$$z_{\text{final}} = \alpha z_c + (1 - \alpha) z_\theta, 0 \leq \alpha \leq 1.$$

- ▶ At early epochs, $\alpha \approx 0$, so the model behaves like a standard DeepONet.
- ▶ At late epochs, $\alpha \rightarrow 1$, so the model becomes conservative.

Full architecture

$$(u_0, v_0) \xrightarrow{\text{branch}} b_k(u_0, v_0)$$

$$\phi \xrightarrow{\text{trunk}} T_k(\phi)$$

$$(b_k, T_k) \xrightarrow{\text{contraction}} z_\theta(\phi; u_0; v_0)$$

$$z_\theta \xrightarrow{\text{projection}} z_c \in \mathcal{M}_{I_0}.$$

See <https://arxiv.org/abs/2311.14131>.

|| VL via PINN ||

- 🔧 Learns the trajectory
- 🔧 Learns Structure only weakly imposed

|| cPINNs and cDeepONets ||

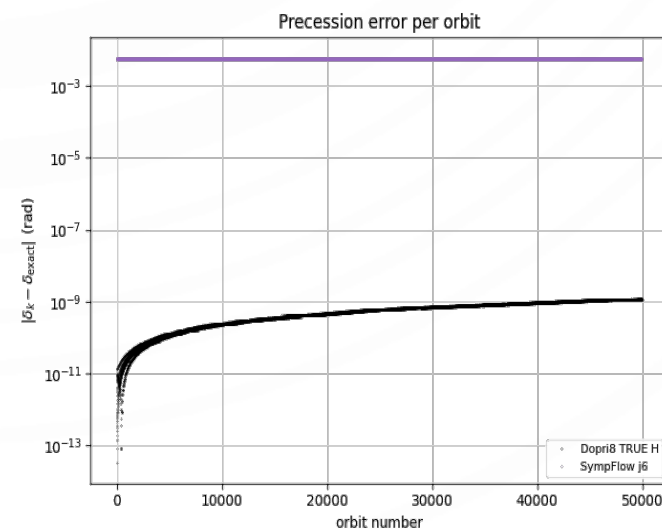
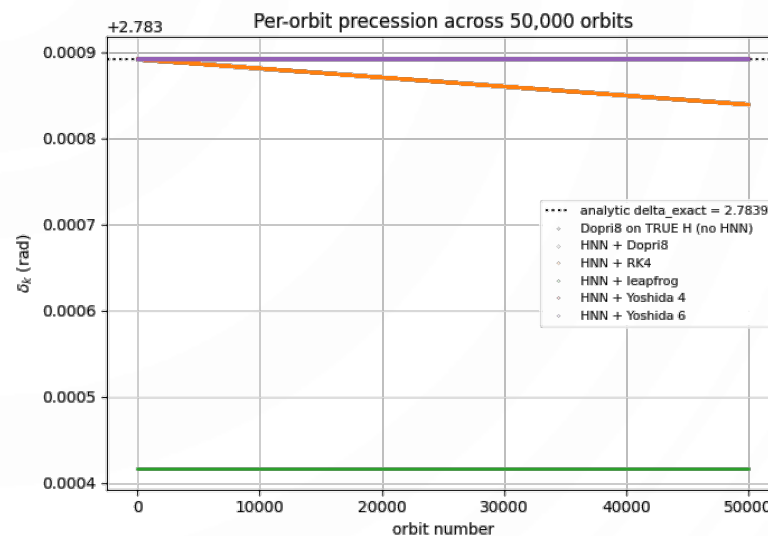
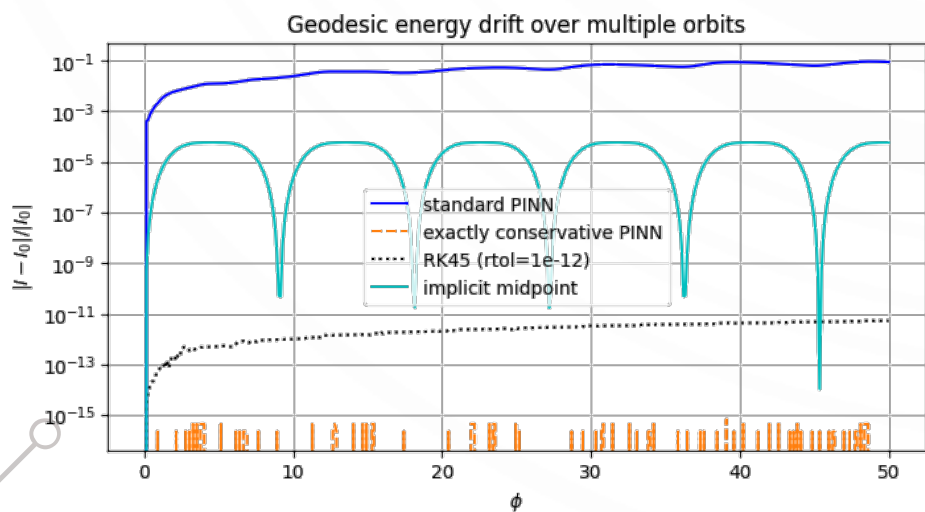
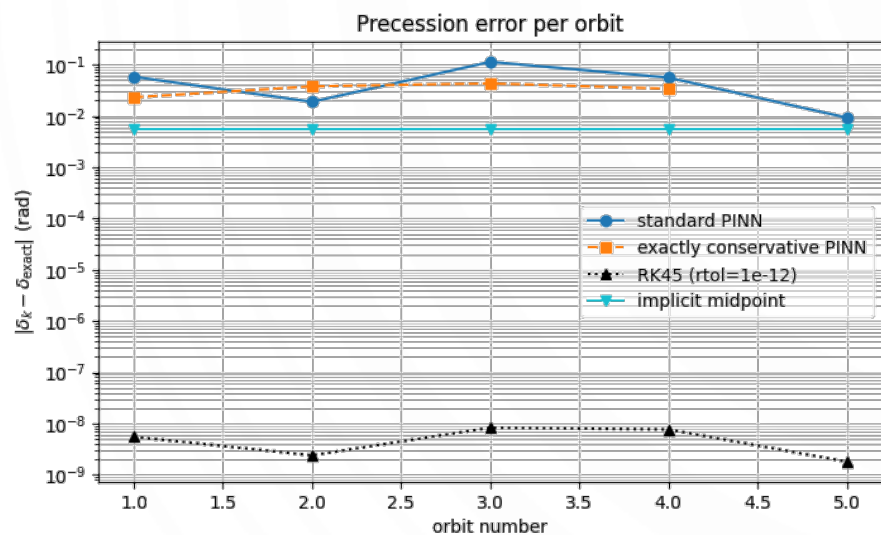
- 🔧 Learns a state and corrects it
- 🔧 Structure is imposed only on the invariant manifold via brute-force

|| VL via HNN ||

- 🔧 Learns the Hamiltonian
- 🔧 Vector field is Hamiltonian by construction.

|| SympFlow / flow-map method ||

- 🔧 Learns the evolution map with a symplectic constraint
- 🔧 Time-step itself preserves phase-space structure.



They inject stronger geometric inductive bias.

» The observable is not just the local trajectory error. For relativistic orbits, the important quantity is long-time phase.

Invariant error: $\epsilon_I(\phi) = \left| \frac{I(z_\theta(\phi)) - I_0}{I_0} \right|$

Measured precession: $\delta_k = \phi_{\text{peri}}^{(k+1)} - \phi_{\text{peri}}^{(k)} - 2\pi$

The exact relativistic precession is: $\delta_{\text{exact}} = \Delta\phi_{\text{rad}} - 2\pi$.

So the precession error is: $\epsilon_{\text{prec}}^{(k)} = |\delta_k - \delta_{\text{exact}}|$.

Main causal chain:

► $I(z) \neq I_0$ ► wrong invariant curve in phase space ► wrong radial period ► wrong accumulated periapsis phase

For GW dynamics, long-time phase accuracy IS A GEOMETRIC PROBLEM.

|| GNML || INTRODUCING SINFONIA-J ||

What is **SINFONIA**?



Symplectic Neural Flows for Orbital Numerical Integration and Acceleration

SINFONIA-J0

🔧 Learns full symplectic map

Example: OOD from 3PN dynamics

Again, what do we want ?

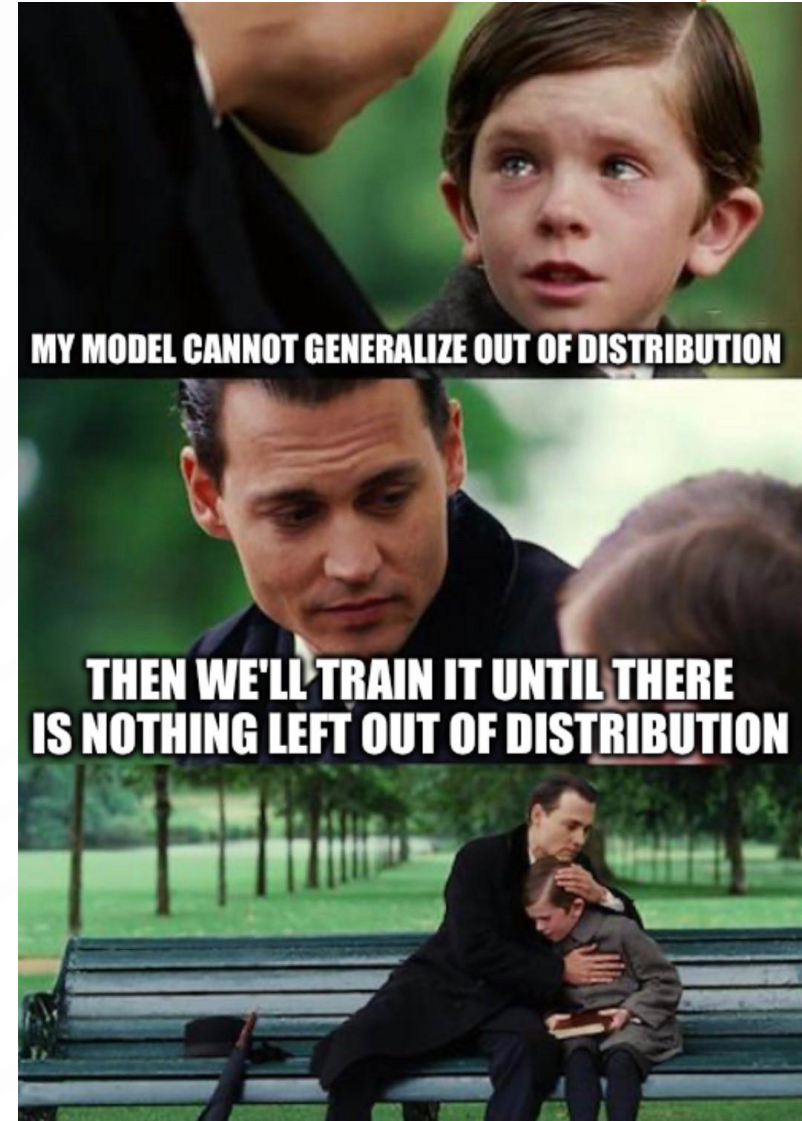
- ⚙️ Structure-preserving geometric methods
- ⚙️ Long-term stability
- ⚙️ Amortization across series of initial conditions
- ⚙️ Out of distribution computational power

L.J. GOMES DA SILVA

SINFONIA-JI

🔧 It is going to learn a ghost symplectic neural flow map

Example: Inspiral from NS-NS binary



|| GNML || SINFONIA-JO :::: THE PHYSICAL PROBLEM ||

⚙ The Hamiltonian:

$$H_{\text{orb}}^{(p)}(q, p) = H_N + \sum_{k=1}^p H_{k\text{PN}}$$

$$H_N = \frac{|p|^2}{2\mu} - \frac{GM\mu}{r}, \quad \mu = \frac{m_1 m_2}{m_1 + m_2}, \quad M = m_1 + m_2$$

⚙ Dynamics of the Hamiltonian:

$$\dot{z} = J\nabla H(z),$$

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

⚙ Train $\Phi_\theta(z, t)$:

$$\Phi_\theta(z, t) \approx \Phi_t^{H^{(p)}_{\text{orb}}}(z)$$

🔧 Symplectic by construction

🔧 A single network should handle any time step $t \in [-\Delta t, \Delta t]$.

🔧 Rollout is just repeated function evaluation at:

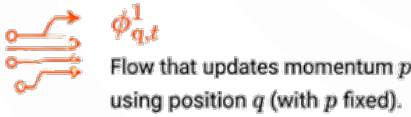
$$z_{n+1} = \Phi_\theta^h(z_n, \Delta t)$$

|| GNML || SINFONIA-JD :::: ARCHITECTURE IN A NUTSHELL ::: SYMPFLOW ||

- ➡ A time-dependent neural flow layer is associated with the Hamiltonian:

$$H_t^i(q, p) = V_p^i(t, p) + V_q^i\left(t, q - \left(\nabla_p V_p^i(t, p) - \nabla_p V_p^i(t, 0)\right)\right).$$

- ➡ Each layer is implemented as alternating exact Hamiltonian subflows:

$$p \leftarrow p - \left(\nabla_q V_q^{(i)}(q, t) - \nabla_q V_q^{(i)}(q, 0)\right)$$


Flow that updates momentum p using position q (with p fixed).

$$q \leftarrow q + \left(\nabla_p V_p^{(i)}(p, t) - \nabla_p V_p^{(i)}(p, 0)\right)$$


Flow that updates position q using momentum p (with q fixed).

- ➡ Because SympFlow is a composition of Hamiltonian maps, the total Hamiltonian is built recursively:

$$H_t^{L:i}(x) = H_t^{L:(i+1)}(x) + H_t^i\left(\phi_{H_t^{L:(i+1)},t}^{-1}(x)\right), i = 1, \dots, L - 1,$$

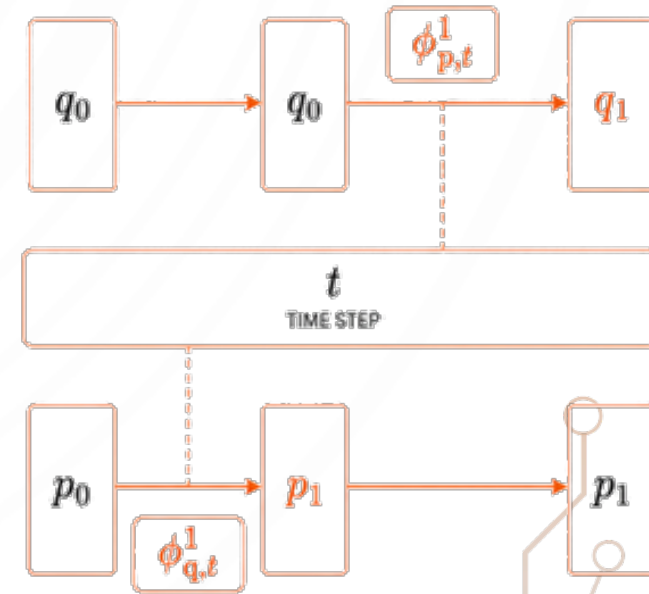
with $H_t^{L:L} = H_t^L$. Therefore, the Hamiltonian associated with the whole SympFlow is $\mathcal{H}(\psi) = H_t^{L:1}$.

- ➡ Full network:

$$\Phi_\theta^h = \Phi_{\theta_L}^{h,L} \circ \dots \circ \Phi_{\theta_1}^{h,1}$$

Composition of symplectic maps is symplectic, then $\Phi_\theta(z, t)$, is symplectic for any parameter setting!

Each of i layers is the exact-flow of the separable time-dependent Hamiltonian



<https://arxiv.org/pdf/2412.16787>

|| GNML || SINFONIA-JO :::: ARCHITECTURE :::: TRAINING LOSSES ||

|| Residual (PINN-like) ||

$$L_{\text{res}} = \frac{1}{N} \sum_i \left\| \partial_t \Phi_{\theta}(z_i, t_i) - J \nabla H(\Phi_{\theta}(z_i, t_i)) \right\|^2$$

|| H-match (anchors the learned rollout) ||

$$L_H = \frac{1}{N} \sum_i (H_{\theta}(z_i, t_i) - H_{\text{true}}(z_i))^2$$

|| Supervised (in our case SPN4 trajectories) ||

$$L_{\text{sup}} = \frac{1}{N} \sum_i \left\| \Phi_{\theta}(z_i, t_i) - z_i^{\text{SPN4}}(t_i) \right\|^2$$

💀 || Normalization matters !!! || 💀

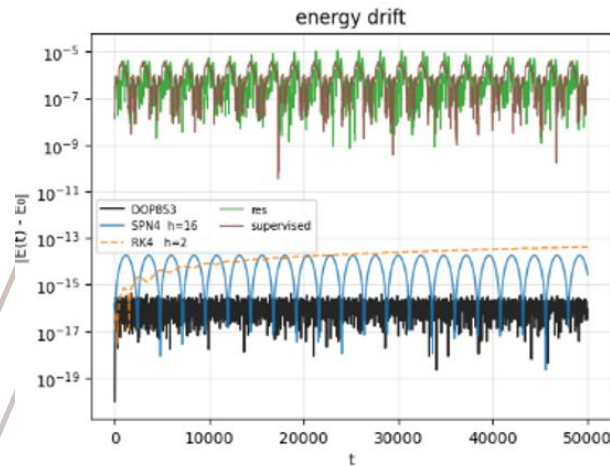
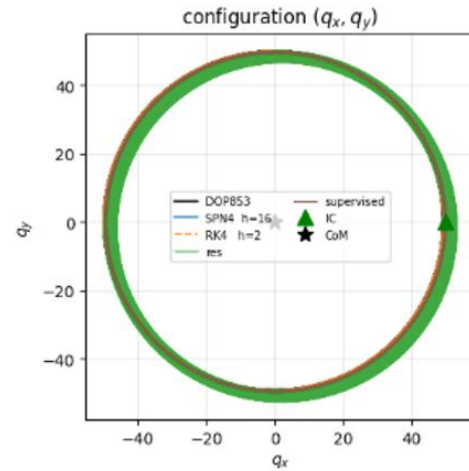
The positions are of order $q \sim 50$, while the momenta are of order $p \sim 0.027 \rightarrow$ tahn saturates @ some range

Without normalization :::: network fights scale before it fights dynamics

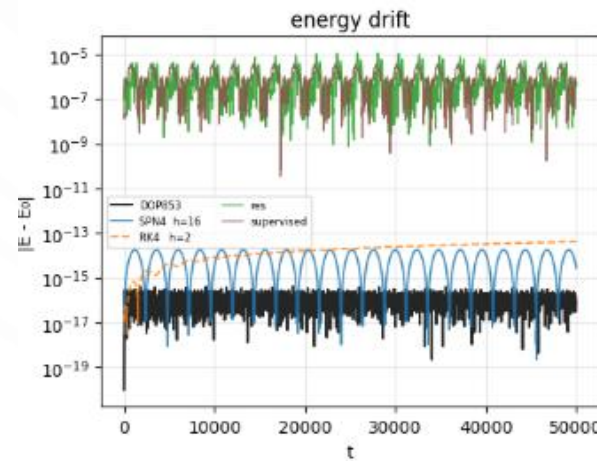
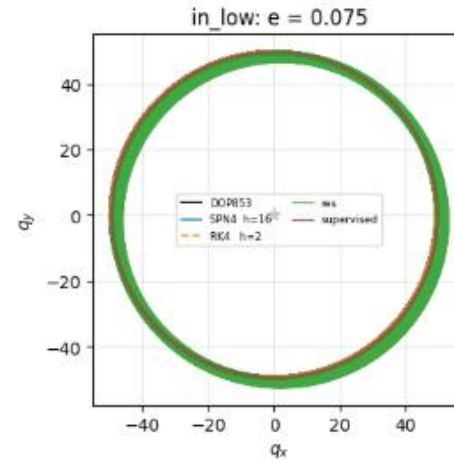
|| GNML || SINFONIA-JO || CONSERVATIVE DYNAMICS @ 3PN || RESULTS

Multi-IC training @@@ Data range = {0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60}

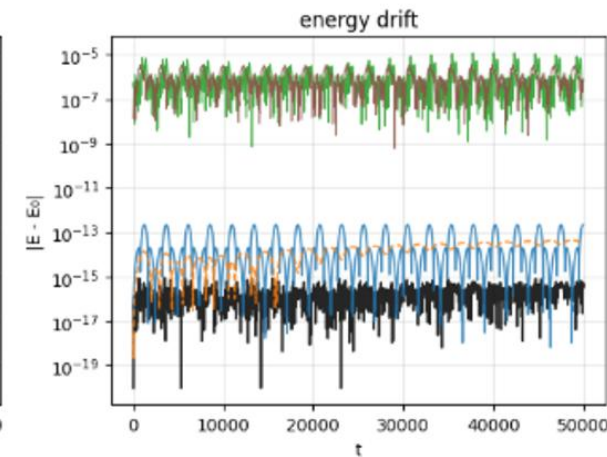
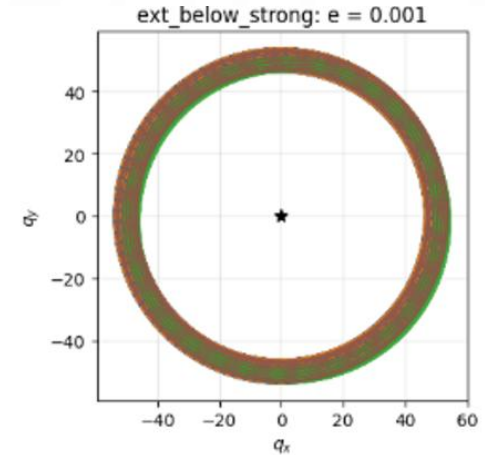
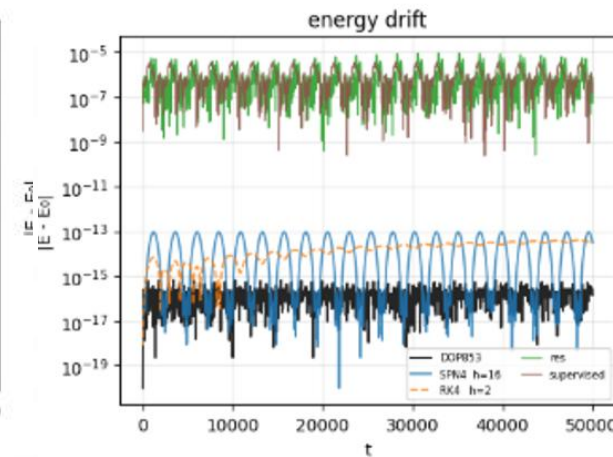
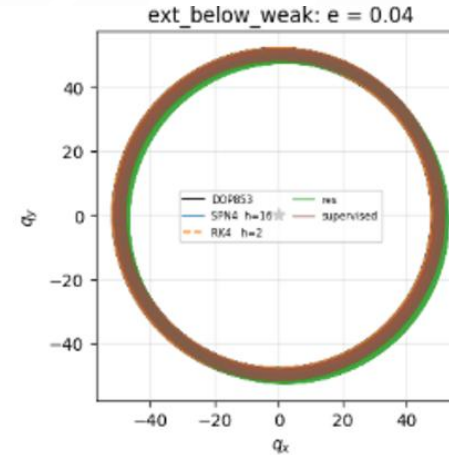
In range:



Interpolated within training data set:



Weak OOD: Extrapolated value



⚙️ **Source:** Compact binary inspiral, 2.5PN relative motion

⚙️ Relative coordinate $\mathbf{q}$, mass ratio ν , total mass M .

⚙️ Inspiral from $r_0 = 100 M$ to ISCO at $r = 6 M$.

⚙️ For an equal-mass NS-NS binary $t_{\text{coal}} \approx 108 \text{ s}$, ~ 2000 orbits, accumulated phase $\sim 12,500 \text{ rad}$.

⚙️ Conservative Lagrangian:

$$L = \frac{1}{2} \mu |\dot{\mathbf{q}}|^2 + \frac{\mu M}{|\mathbf{q}|}$$

⚙️ At 2.5PN, radiation reaction enters as a non-conservative force:

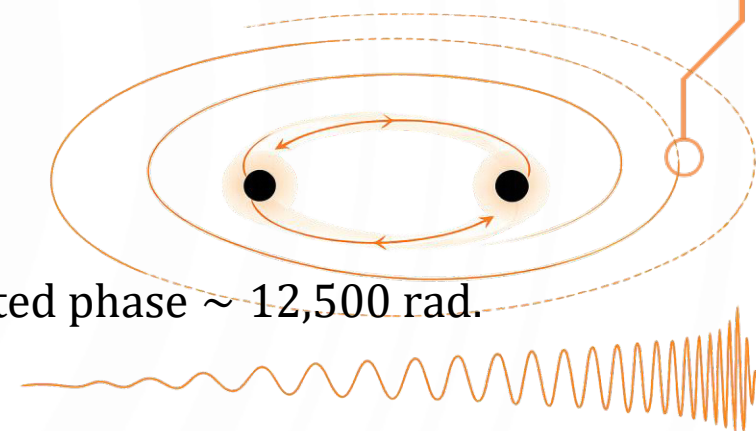
$$\mu \ddot{\mathbf{q}} = \mathbf{F}_{\text{cons}} + \mathbf{F}_{\text{RR}}^{2.5\text{PN}}.$$

💡 As previously discussed, we use Galley's idea: restore a variational description by doubling variables,

$$q \rightarrow (q_+, q_-),$$

and generating dissipation through a non-conservative potential K .

INSPIRALLING BINARY (PN DYNAMICS)



A Taylor-informed ansatz is

$$\Phi_{\theta}(u, t) = u + \sum_{k=1}^p \frac{t^k}{k!} f_k(u) + t^{p+1} \text{MLP}_{\theta}(u, t).$$

Here,

- $f_k(u)$ are the **analytic Taylor coefficients of the true flow at $t = 0$**
- The term $\text{MLP}_{\theta}(u, t)$ is a learned remainder. It only learns what is missing beyond the analytic Taylor expansion.
- The loss compares the learned flow to one step of a reference symplectic integrator:

$$\mathcal{L}_{\text{res}} = \left\| \Phi_{\theta}(u, t + h) - \Phi_h^{\text{slimp}}(\Phi_{\theta}(u, t)) \right\|^2.$$

💀 ENFORCE STRONG GEOMETRIC BIAS 💀

Instead of using a completely unconstrained MLP remainder, we can build the learned correction from **gradient-symplectic layers**.

An **up layer** updates momentum while keeping position fixed:

$$\text{up}: (q, p) \mapsto (q, p + \nabla_q F_i(q, t)).$$

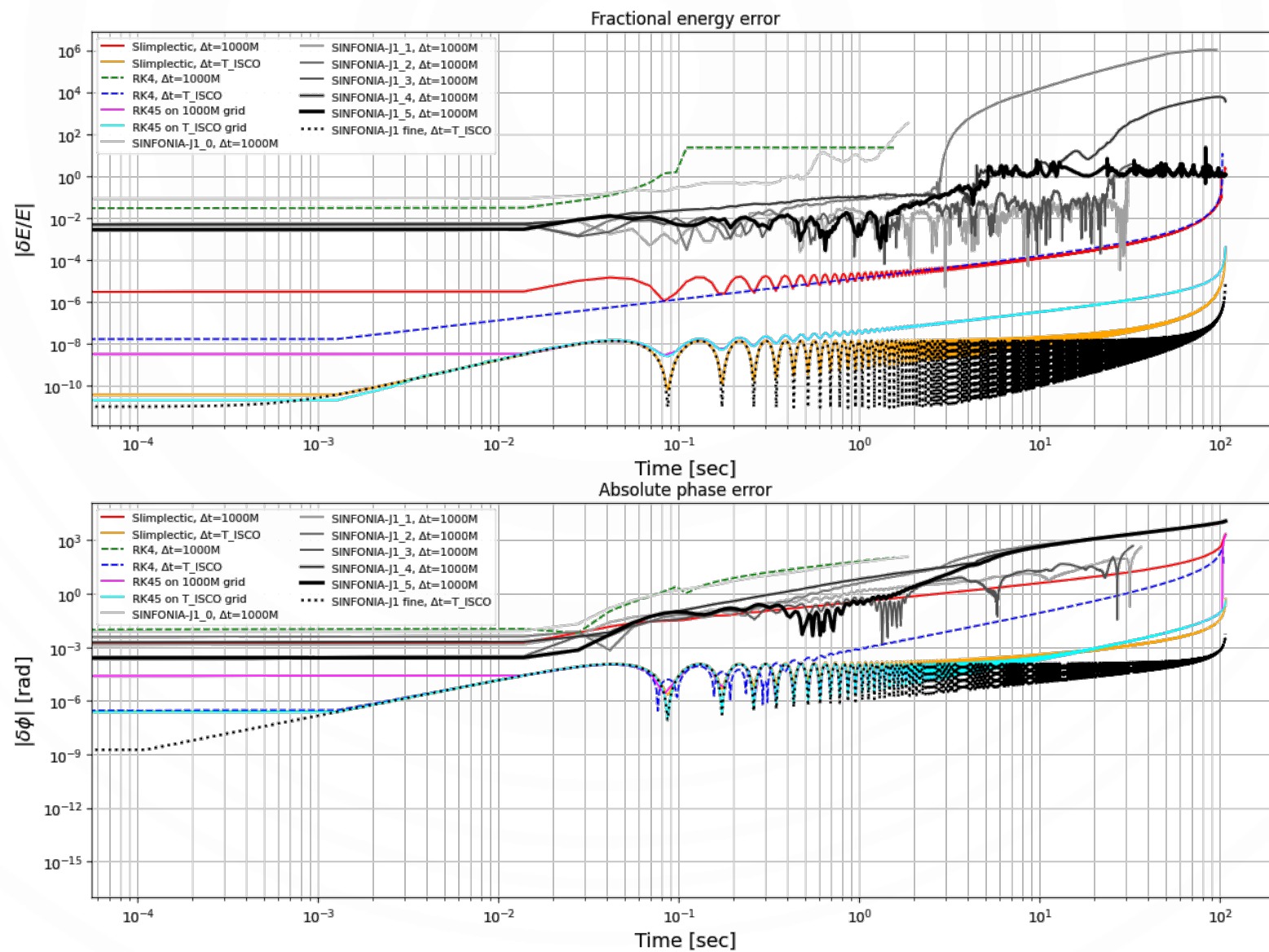
A **down layer** updates position while keeping momentum fixed:

$$\text{down}: (q, p) \mapsto (q + \nabla_p G_i(p, t), p).$$

The learned map is written as

$$\Phi_{\theta}(u, t) = \underbrace{\text{SympNet}(u, t)}_{\text{conservative core}} + \underbrace{\left(0, t \varepsilon_K \left. \frac{\partial K_{\theta}}{\partial q_-} \right|_{q_- = v_- = 0} \right)}_{\text{Galley \(\kappa\) dissipation}} + \underbrace{t \varepsilon_{\text{aux}} \text{MLP}_{\text{aux}}}_{\text{stabilizer}}.$$

|| GNML || SINFONIA-J1 :::: RESULTS ||



|| OUTLOOK ||

- 📺 Improvement of SINFONIA-J0 architectural layer design and push to more challenging OOD extrapolation regions;
- 📺 Benchmark both SINFONIA-J0 and SINFONIA-J1 at large scale against DOP853, including the amortized versions and compare long-term energy drift and speeds;
- 📺 Add a symbolic regression branch to SINFONIA-J0 and SINFONIA-J1:
 - 🍏 If trained supervised with some simulated dataset from a model incorporated with environmental effects or beyond GR corrections, can SINFONIA:
 - integrate OOD, evolve with no phase drift?
 - retrieve the correct analytical expression via symbolic regression?
- 📺 Extend framework to EOB and train on simulated IMR data from both BOB and NR data;

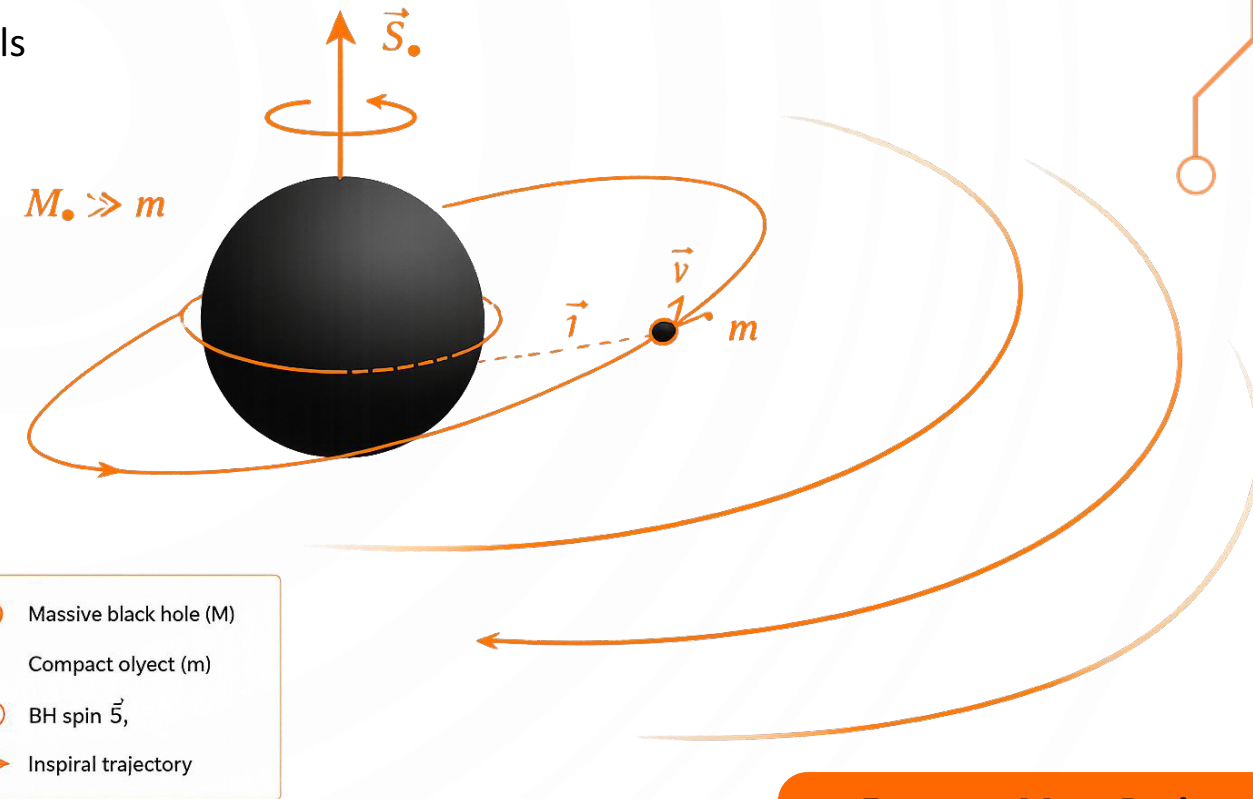
|| RESERVE SLIDES ||

|| GSF AND BHPT IN A NUTSHELL :::: MODELLING AND E(X)MRI ||

⚙ Sources: E/(X)MRIs Extreme-(eXtremely)- Mass ratio inspirals

|| Key features ||

- ⚙ [F(eature) 1]: Extreme length scales disparity
- ⚙ [F2]: Very slow, $\epsilon^{-1} \approx 10^4 - 10^7$ orbits
- ⚙ [F3]: Weak amplitudes but high SNR of GW emission & detection
- ⚙ [F4]: Generically eccentric & spinning & inclined
- ⚙ [Alt F4]: Circular &/or equatorial orbits
- ⚙ [F5]: Complex IMR with sudden transition from inspiral to plunge
- ⚙ [F6]: EMRBs emission from scattering/hyperbolic encounters



Extreme Mass Ratio
 $\epsilon = \mu/M$
 $[\epsilon \approx 10^{-6/-7} - 10^{-4}]$

|| GSF AND BHPT IN A NUTSHELL :::: A SKETCH OF THE NUMERICAL RECIPE ||

➡ Expanding the Einstein tensor to linear order:

$$G_{\alpha\beta} = \bar{G}_{\alpha\beta} + \delta G_{\alpha\beta} \rightarrow \delta G_{\alpha\beta} = \frac{8\pi G}{c^4} T_{\alpha\beta}$$

$$T_{\alpha\beta} = \mu \int d\tau (-g)^{1/2} u_\alpha u_\beta \delta^4(x^\mu - z^\mu(\tau))$$

➡ Split **4D** spacetime $x^\alpha = (x^a, \theta^A)$:

2D Lorentzian manifold $x^a = (t, r)$ + **2D** Sphere $\theta^A = (\theta, \phi)$

➡ Apply scalar **OR** tensor spherical harmonic decomposition s.t.

🔧 Scalar

$$\Phi(t, r, \theta, \phi) = \sum_{l=0}^{\infty} \phi(t, r) Y_{lm}(\theta, \phi)$$

➡ Apply **Klein Gordon** eq. for a scalar field

$$\square \Phi = \nabla_\mu \nabla^\mu \Phi = \frac{1}{\sqrt{-g}} \partial_\mu [\sqrt{-g} g^{\mu\nu} \partial_\nu \Phi] = S_\Phi$$

where,

$$S_\Phi = -4\pi q \int_{-\infty}^{\infty} \frac{1}{\sqrt{-g}} \delta^4(x^\mu - z_p^\mu(\tau)) d\tau$$

➡ Simplify to “amicable” form

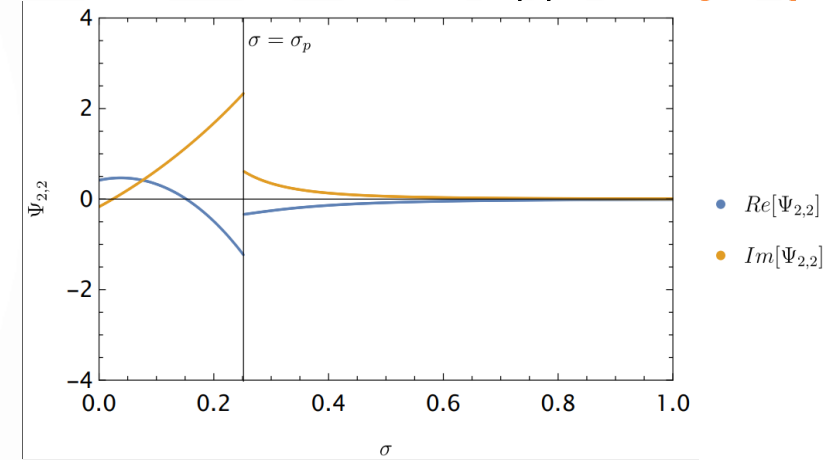


Figure 2 - From [DiscoTEX P1](#)

The numerical recipe for solving governing PDE:::

$$\begin{aligned} [-\partial_t^2 + \partial_x^2 - V^{S/RW/Z}(r)] \phi^{S/RW/Z}(t, r) &= S^{S/RW/Z}(t, r) \\ S^{S/RW/Z}(t, r) &= G(t, r) \delta(r - r_p(t)) + I(t, r) \delta'(r - r_p(t)) \end{aligned}$$

Then goes as

⚙️ **Phase 1:** Resolve the wave-like equation and compute master functions

$$\phi^{S/RW/Z}(t, r) \text{ \& } \partial_t \phi^{S/RW/Z}(t, r)$$

⚙️ **Phase 2:** Reconstruct the metric perturbations || We've 6 perturbations

$$h(t, r) = \beta^{lm}(t, r) = [f \phi^{RW} + r f \partial_r \phi^{RW} - r^3 E_c] \text{ @ particle i.e } \beta^\pm(t, r_p)$$

⚙️ **Phase 3:** Use reconstructed terms to calculate the GSF

$$F_{RW, odd}^{r, lm} = [\beta^{lm}(t, r) + r_p \partial_r \beta^{lm}(t, r)] \partial_\theta Y_{lm} \left(\frac{\pi}{2}, \Omega t \right) \text{ @ particle i.e } F_{RW, odd}^{r, lm, \pm}(t, r_p)$$

|| GSF AND BHPT IN A NUTSHELL ::: NUMERICAL RECIPE VIA DISCOTEX ||

$$[-\partial_t^2 + \partial_x^2 - V^{S/RW/Z}(r)]\phi^{S/RW/Z}(t, r) = S^{S/RW/Z}(t, r) = G(t, r)\delta(r - r_p(t)) + I(t, r)\delta'(r - r_p(t))$$

DiscoTEX 1.0 & Co - Philosophy

⚙️ Difficult 1 in space - Disco

🔧 Discontinuous collocation methods for **spatial discretisation** (Discretize U, L in space using finite difference, pseudospectral or Fourier collocation nodes)

Example:

Chebyshev/equidistant collocation nodes

$$L \text{ is simply } \Leftrightarrow \begin{pmatrix} 0 & I \\ \partial_x^2 & 0 \end{pmatrix}$$

⚙️ Difficulty 1 in time - Disco

🔧 Discontinuous Hermite/RK interpolation formulae

$$U^{n+1} = \left(I - \frac{\Delta t}{2} L + \frac{\Delta t^2}{12} L^2 + \dots \right)^{-1} \cdot \left(I + \frac{\Delta t}{2} L \frac{\Delta t^2}{12} L^2 + \dots \right) U^n$$

⚙️ Difficulty 3 - TEX/IMP/REX

🔧 Use **IMTEX** implicit-turned-explicit integration scheme. (Integrate in time using time symmetric methods as a coupled ODE system).

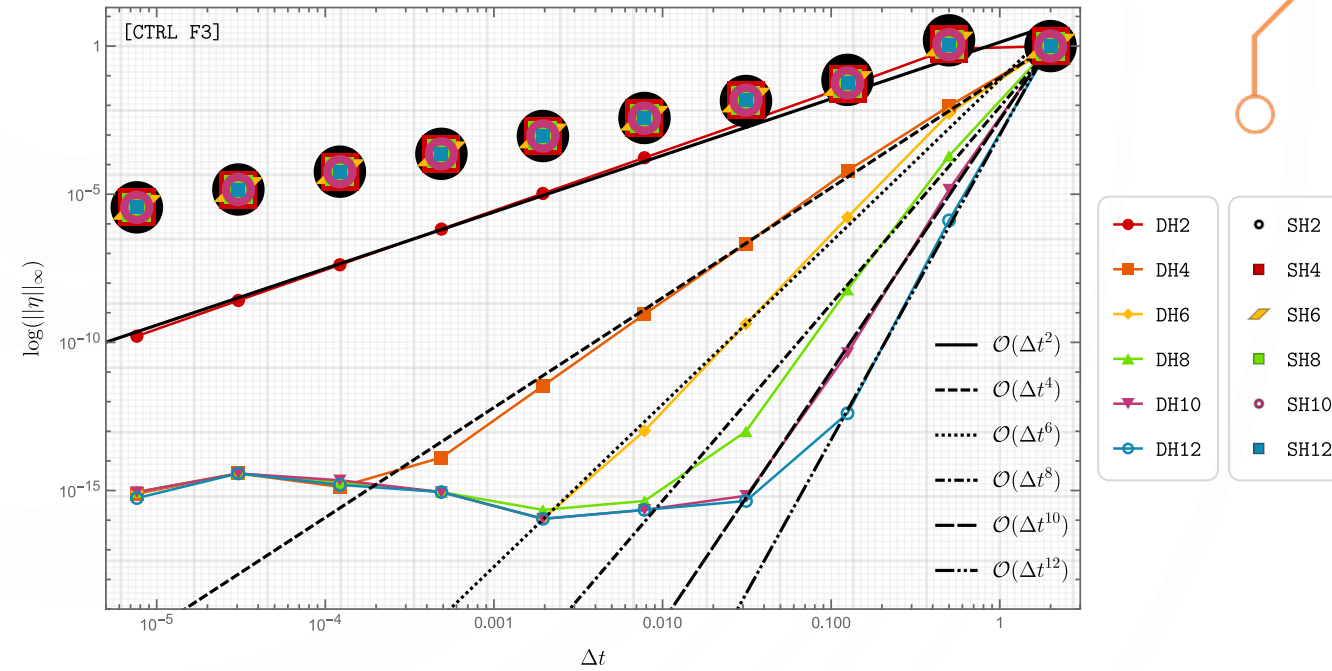


Figure 3 - From [DiscoTEX P2](#)

🔧 The DiscoTEX Algorithm 🔧

$$U^{n+1} = U^n + M. \left(L. (\Delta t U^n) - (r) + \frac{\Delta t}{2} (S^n + S^{n+1}) + \llbracket u \rrbracket \right)$$

- ⚙️ Difficult 1 [F1, F3] – Handling the distributional source term in both space and time, 1+1D
- ⚙️ Difficult 3 [F2, F3] – EMRIs are slow evolution systems, their evolution may take months or even years!

On the Stability of the Iterated Crank-Nicholson Method in Numerical Relativity

Saul A. Teukolsky

Newman Laboratory, Cornell University, Ithaca, NY 14853

(November 26, 2024)

In the late 90's** implicit methods start being explored through the **Iterated Crank Nicholson** (ICN) Scheme written in an explicit form.

Misconceptions seemed to originate here with two major conclusions:

☢ **Proposition I** - The ICN scheme **retains stability for at most two iterations**, any higher number of iterations will result in a less stable method.

☢ **Proposition II** - The accuracy of the ICN scheme remains second order in Δt and Δx from the first iteration on. **Doing more iterations changes the stability behavior, but not the accuracy.** Since the smallest number of iterations for which the method is stable is two, there is no point in carrying out more iterations than this.

We can solve the Iterated Crank-Nicholson Scheme explicitly as:

- **Initializing Step**

$$u^{n+1} = u^n + (\Delta t L).u^n$$

- **Then Iterate k times**

L.J. GOMES DA SILVA

$$u^{n+1} = u^n + \frac{\Delta t}{2} L. (u^{n+1} + u^n)$$

$$U_{n+1}^{(k+1)} = \left[1 + \Delta t L \sum_{k=0}^{k_{max}-1} \left(\frac{\Delta t L}{2} \right)^k \right] U_n$$

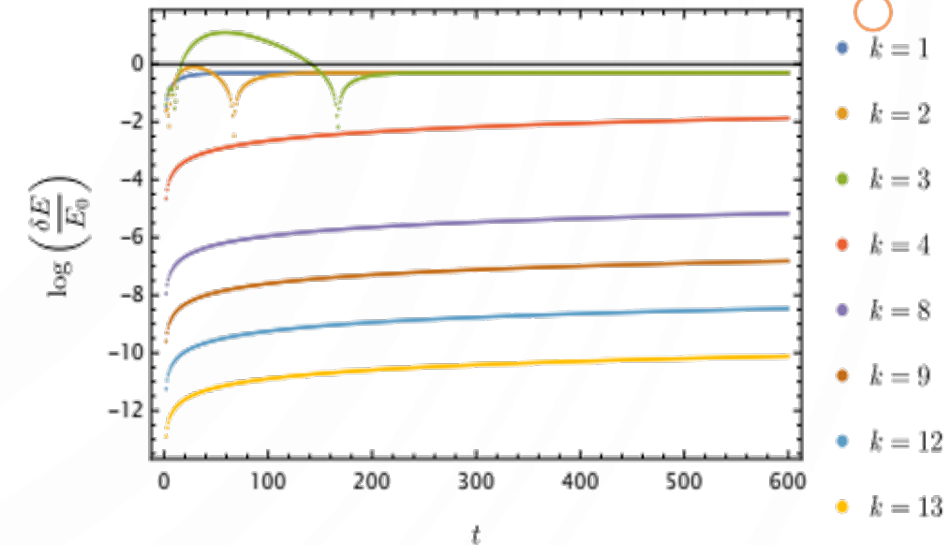


Figure 4 – Increasing the number of k iterations effectively changing the accuracy of the ICN. <https://arxiv.org/abs/gr-qc/9909026>. Plot from my PhD Thesis.

|| GSF AND BHPT IN A NUTSHELL || | FOCUSING ON TIME-INTEGRATION ||

⚙ Consider the ODE in time:

$$\frac{dy}{dt} = f(y, t)$$

🔧 Fundamental theorem of calculus:

$$y(t_2) - y(t_1) = \int_{t_1}^{t_2} f(y, t) dt$$

$$\begin{aligned} y(t_2) - y(t_1) &\approx \\ \frac{\Delta t}{2} (f_1 + f_2) + \frac{\Delta t^2}{10} (f'_1 - f'_2) + \frac{\Delta t^3}{120} (f''_1 - f''_2) + J(\Delta t_x, \Delta t) \end{aligned}$$

🔧 For PDEs:

$$U(t_{n+1}) = U(t_n) + \int_{t_n}^{t_{n+1}} LU(t) dt$$

2-point Taylor expansion

Hermite order 2 aka trapezium rule

$$U_{n+1} = U_n + \frac{\Delta t}{2} L. (U_n + U_{n+1}) + \mathcal{O}(\Delta t^3)$$

IMTEX Hermite order 4

$$= \left(I - \frac{\Delta t}{2} L + \frac{\Delta t^2}{12} L^2 + \dots \right)^{-1} \cdot \left(I + \frac{\Delta t}{2} L + \frac{\Delta t^2}{12} L^2 + \dots \right) U_n$$

Hermite order 4

$$U_{n+1} = U_n + \frac{\Delta t}{2} L. (U_n + U_{n+1}) + \frac{\Delta t^2}{12} L. (\dot{U}_n - \dot{U}_{n+1}) + \mathcal{O}(\Delta t^5)$$

|| BUT WHAT IS SYMPLECTIC STRUCTURE ??? WHAT DO WE MEAN BY PRESERVATION ? ||

Considering the Hamiltonian

$$\dot{q} = \frac{dq}{dt} = \frac{\partial H}{\partial p} = \nabla_p H \quad \dot{p} = \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\nabla_q H \quad \frac{d}{dt} \begin{bmatrix} q \\ p \end{bmatrix} = J^{-1} \nabla H(q, p) \quad u = J^{-1} \nabla H(u) \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

Flow of the system

$$\varphi_t: (q(0), p(0)) \rightarrow (q(t), p(t)) \quad \varphi_t: \mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$$

Energy preserving time-integrator

$$\frac{q_{n+1} - q_n}{\Delta t} = \frac{H(q_n, p_{n+1}) - H(q_n, p_n)}{p_{n+1} - p_n}$$

$$\frac{p_{n+1} - p_n}{\Delta t} = -\frac{H(q_n, p_{n+1}) - H(q_{n+1}, p_{n+1})}{q_{n+1} - q_n}$$

$$H(q_{n+1}, p_{n+1}) - H(q_n, p_n) = 0$$

Symplectic time-integrator

$$u_h: (q_n, p_n) \rightarrow (q_{n+1}, p_{n+1}) \quad u_h: \mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$$

$$U_h := \begin{bmatrix} \frac{\partial u_h(q_n, p_n)}{\partial q_n} & \frac{\partial u_h(q_n, p_n)}{\partial p_n} \end{bmatrix}$$

$$U_h^T J U_h = J$$

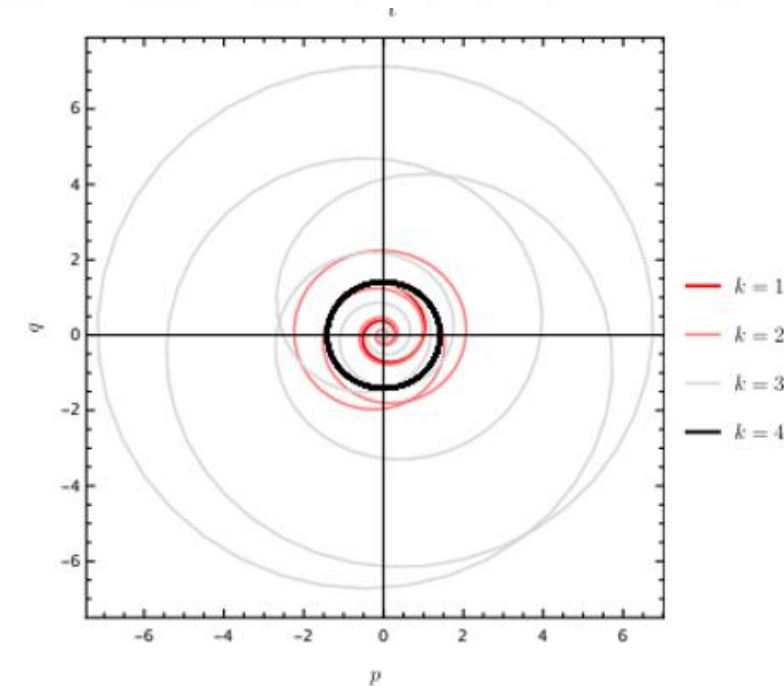


Figure 5 – Increasing the number of k iterations effectively changing the accuracy of the IMTEX ICN and allowing the recovery of symplectic structure preservation <https://arxiv.org/abs/gr-qc/9909026>. Plot from my PhD Thesis. See Appendix B of [DiscoTEX P1](#).

|| BUT WHAT IS SYMPLECTIC STRUCTURE ??? || EXPLICIT DEMONSTRATION ||

Consider the simple harmonic oscillator :

$$H(p, q) = \frac{1}{2}(p^2 + q^2)$$

$$\frac{dU}{dt} = LU$$

$$U = \begin{pmatrix} p \\ q \end{pmatrix}$$

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

Is the IMTEX H2 scheme energy-preserving?

$$\begin{pmatrix} q_{n+1} \\ p_{n+1} \end{pmatrix} = \begin{pmatrix} -1 + \frac{8}{4 + \Delta t^2} & \frac{4\Delta t}{4 + \Delta t^2} \\ -\frac{4\Delta t}{4 + \Delta t^2} & -1 + \frac{8}{4 + \Delta t^2} \end{pmatrix} \begin{pmatrix} q_n \\ p_n \end{pmatrix}$$

$$\frac{1}{\Delta t} \left[-2 + \frac{8}{4 + \Delta t^2} \right] q_n + \frac{4\Delta t}{4 + \Delta t^2} p_n = \frac{H(p_{n+1}, q_n) - H(p_n, q_n)}{p_{n+1} - p_n}$$

$$\frac{1}{\Delta t} \left[-2 + \frac{8}{4 + \Delta t^2} \right] p_n - \frac{4\Delta t}{4 + \Delta t^2} q_n = \frac{H(p_{n+1}, q_n) - H(p_{n+1}, q_{n+1})}{q_{n+1} - q_n}$$

$$\frac{1}{\Delta t} c_1^2 p_n q_n - c_2 c_1 q_n^2 + c_2 c_1 p_n^2 - \Delta t c_2^2 p_n q_n = H(p_{n+1}, q_n) - H(p_n, q_n) \quad \frac{1}{\Delta t} c_1^2 p_n q_n - c_2 c_1 q_n^2 + c_2 c_1 p_n^2 - \Delta t c_2^2 p_n q_n = H(p_{n+1}, q_n) - H(p_{n+1}, q_{n+1})$$

Is it symplectic?

$$\det U_{\text{IMTEX HF2}} = \left(-1 + \frac{8}{4 + \Delta t^2} \right)^2 + \left(\frac{4\Delta t}{4 + \Delta t^2} \right)^2 = 1$$

$$U_h^T J U_h = \begin{pmatrix} -1 + \frac{8}{4 + \Delta t^2} & \frac{\Delta t}{1 + \Delta t^2/4} \\ -\frac{\Delta t}{1 + \Delta t^2/4} & -1 + \frac{4}{4 + \Delta t^2} \end{pmatrix} J \cdot \begin{pmatrix} -1 + \frac{8}{4 + \Delta t^2} & -\frac{\Delta t}{1 + \Delta t^2/4} \\ \frac{\Delta t}{1 + \Delta t^2/4} & -1 + \frac{4}{4 + \Delta t^2} \end{pmatrix}$$

|| OUTLOOK ||

- 📺 Improvement of SINFONIA-J0 architectural layer design;
- 📺 Benchmark both SINFONIA-J0 and SINFONIA-J1 at large scale against DOP583, including the amortized versions and compare long-term energy drift and speeds;
- 📺 Add a symbolic regression branch to SINFONIA-J0 and SINFONIA-J1
 - 🍏 If trained supervised with some simulated dataset from a model incorporated with environmental effects or beyond GR corrections, can SINFONIA:
 - integrate OOD, evolve with no phase drift?
 - retrieve the correct analytical expression via symbolic regression?
- 📺 Extend framework to EOB and train on simulated IMR data from both BOB and NR data;