

# Exact and penalized physical constraints in engineering neural field models

Luis Torres

## Abstract

Physical constraints can enter a neural field model through its architecture, its parameterization, or its optimization objective. These mechanisms have different mathematical meanings. An algebraic boundary ansatz can enforce a prescribed trace exactly, while a finite penalty generally permits violation and an augmented-Lagrangian method enforces feasibility only to its achieved tolerance. This review develops those distinctions for elasticity, incompressible flow, and heat transfer. Analytical examples show how exact boundary conditions alter derivatives, how incompatible Neumann data defeat any optimizer, and how penalty strength affects conditioning. Interface laws, constitutive admissibility, uncertain measurements, and viscoplastic internal states are examined as engineering cases where indiscriminate constraint enforcement can be misleading. A proposed verification protocol evaluates both the constrained quantity and the remaining physical equations.

## 1. What does exact enforcement mean?

Physics-informed neural networks traditionally combine equation, boundary, initial, and observation losses [1, 2]. A finite weighted loss can encourage these requirements without satisfying any one of them exactly. In contrast, a parameterization may restrict every admissible network output to satisfy a particular identity. A two-dimensional streamfunction can enforce incompressibility, and a boundary lifting can enforce specified displacement or temperature values. The distinction is about the set of admissible functions, not whether a paper calls its method hard or soft.

Exactness should always name the constraint and its assumptions. A boundary ansatz may enforce a value only on the geometry used to construct it. A divergence-free representation assumes sufficient smoothness and compatible coordinates. A constitutive potential may impose one derivative relation while leaving convexity or dissipation unconstrained. Numerical floating-point evaluation and approximate geometric representations introduce additional tolerances that should be reported when they affect engineering acceptance.

Distance-function constructions and constrained inverse-design PINNs illustrate different ways of incorporating physical restrictions [3, 4]. They also show why architecture-level identities must be distinguished from optimization-level constraints. Penalty and augmented-Lagrangian iterations can be extremely useful, but their finite-iteration feasibility should be measured rather than presumed. The central engineering question is which physical errors remain possible after a constraint has been imposed.

## 2. Dirichlet lifting: a useful exact construction

Let  $\Gamma_D$  be a boundary where a scalar field  $u$  has prescribed value  $g$ . Suppose  $\tilde{g}$  is an extension into the domain and  $d$  vanishes on  $\Gamma_D$ . A neural correction  $N_\theta$  gives

$$u_\theta(x) = \tilde{g}(x) + d(x) N_\theta(x). \\ u_\theta|_{\Gamma_D} = g, \text{ provided } \tilde{g}|_{\Gamma_D} = g.$$

This statement is algebraic and does not require successful optimization. However, the ansatz alters the derivatives that enter the governing equations. For a twice-differentiable scalar  $d$  and network output  $N$ , the Laplacian is

$$\Delta(dN) = d \Delta N + 2 \nabla d \cdot \nabla N + N \Delta d.$$

The geometry factor can therefore amplify or redistribute derivative terms. If  $d$  is insufficiently smooth at corners or where nearest-boundary distances meet, a strong second-order residual may be unsuitable without a smoother construction or a weak formulation. A product of boundary factors can also become small over an extended region, changing the scale of the neural correction and its gradients. Exact boundary values do not guarantee easy optimization.

On the unit interval, homogeneous endpoint values can be enforced with  $u_\theta(x) = x(1-x)N_\theta(x)$ . Consider the manufactured target  $u(x) = \sin(\pi x)$ , which solves  $-u'' = \pi^2 \sin(\pi x)$ . The exact correction would be  $N(x) = \sin(\pi x)/[x(1-x)]$ . Its endpoint singularities are removable: the limits at both ends equal  $\pi$ . Thus the ansatz can represent this smooth solution without an infinite correction, although implementation should avoid evaluating the quotient naively at the endpoints.

This example demonstrates compatibility of the ansatz with one target solution, not universal approximation for every physical boundary problem. Nonhomogeneous data require a suitable lifting, and mixed boundary conditions require attention to which parts of the boundary should have  $d = 0$ . Multiplying by a factor that vanishes on a traction boundary would impose an unintended displacement condition. The geometric construction must follow the actual boundary partition.

## 3. Compatibility cannot be learned away

Consider the Poisson equation  $-\Delta u = f$  with outward normal derivative  $\partial u / \partial n = g$  on the complete boundary of a bounded domain. Integrating the equation and using the divergence theorem yields the necessary compatibility condition

$$\int_{\Omega} f \, dx = - \int_{\partial \Omega} g \, ds.$$

For a connected domain, a compatible pure-Neumann problem also needs a reference value or zero-mean condition to fix the additive constant. These are distinct issues: the compatibility condition concerns whether a solution exists, while the reference condition concerns uniqueness. Penalizing a pressure or temperature mean cannot repair incompatible source and flux data.

In one dimension,  $-u'' = 1$  on the unit interval with  $u'(0) = u'(1) = 0$  is impossible. Integration gives  $u'(1) - u'(0) = -1$ , contradicting the prescribed derivatives. A soft-constrained optimizer may return

a compromise with nonzero PDE or boundary errors. An exact derivative-boundary parameterization must retain a nonzero PDE residual. Neither outcome is evidence of an inadequate network capacity; the stated physical problem is inconsistent.

The same reasoning is useful in thermal and mechanical engineering. Net heat generation must balance boundary heat flow in a steady problem. Net applied force must be compatible with equilibrium and supports. Thermal PINN and additive-manufacturing applications emphasize the practical importance of specifying sources and boundary losses accurately [5, 6]. Before increasing penalty weights, inspect integral balances and units. A persistent residual can reveal incompatible data rather than a training defect.

#### 4. Penalty strength and conditioning: an exact example

Let  $z$  be a finite-dimensional approximation variable, and consider a quadratic data objective with one linear physical constraint  $a^T z = b$ . The penalized problem is

$$\min_z (1/2) \|z - y\|^2 + (\lambda/2) (a^T z - b)^2, \lambda > 0.$$

$$H = I + \lambda a a^T.$$

For dimension at least two and nonzero  $a$ , the Hessian has eigenvalue 1 in directions orthogonal to  $a$  and eigenvalue  $1 + \lambda \|a\|^2$  along  $a$ . Its Euclidean condition number is therefore  $1 + \lambda \|a\|^2$ . Direct solution gives the constraint violation

$$a^T z_\lambda - b = (a^T y - b) / (1 + \lambda \|a\|^2).$$

Increasing  $\lambda$  reduces violation at order  $1/\lambda$  in this example while making the optimization more anisotropic. The exact constrained minimizer is  $y - a(a^T y - b) / \|a\|^2$ . This comparison isolates the tradeoff without neural-network nonconvexity. It does not prove that every PINN penalty problem has precisely the same conditioning, but it explains why larger weights can improve feasibility while complicating gradient-based optimization.

Units affect  $\lambda$ . Rescaling the constraint by a constant  $c$  changes the effective penalty strength to  $\lambda c^2$  unless the weight is adjusted. Nondimensionalization and residual normalization should therefore precede weight selection. Studies of gradient imbalance, neural-tangent-kernel behavior, and PINN failure modes establish that optimization interactions are already substantial in practical networks [7, 8, 9]. Self-adaptive and relative-progress balancing methods address weight selection, but they do not turn finite penalties into algebraic constraints [10, 11].

An augmented Lagrangian adds a multiplier term and updates the multiplier in conjunction with a penalty. It can improve enforcement without requiring one enormous fixed penalty, subject to algorithmic assumptions and successful subproblem solutions. At any finite iteration, report primal constraint violation and relevant optimization residuals. The constrained inverse-design work of Lu et al. is an example where such mechanisms are integrated into physics-informed optimization [4]. Calling the overall method hard-constrained does not eliminate the need to specify which restrictions are exact by construction and which are solved numerically.

## 5. Divergence-free flow is only part of the problem

For a sufficiently smooth two-dimensional streamfunction  $\psi$ , define  $u = \partial\psi/\partial y$  and  $v = -\partial\psi/\partial x$ . Commutation of mixed partial derivatives makes  $\partial u/\partial x + \partial v/\partial y$  identically zero. Momentum balance remains to be enforced. Substitution into a viscous primitive-variable momentum residual can require third derivatives of  $\psi$ , so an exact conservation identity may increase derivative cost elsewhere.

No-slip wall conditions are not automatically satisfied merely because velocity is divergence-free. The streamfunction and its appropriate derivatives must encode the desired normal and tangential velocities. Nontrivial topology and flux conditions can also affect the representation. A pressure reference may still be necessary because incompressible momentum often determines pressure only up to a constant. These requirements should be listed separately from continuity.

Mixed-variable flow PINNs explore alternative output and residual choices [12]. Hidden-fluid and vortex-induced-vibration studies show how physical equations can support reconstruction from restricted observations [13, 14]. The broader fluid-mechanics literature emphasizes the range of formulations and application conditions [15]. None implies that exact incompressibility alone guarantees accurate pressure drop, drag, or wake dynamics. Independent momentum, traction, and phase diagnostics remain necessary.

## 6. Interfaces and constitutive admissibility

At a perfectly bonded interface between two conducting materials with no interfacial source, temperature and normal heat flux are continuous. The temperature gradient generally jumps when conductivity changes. A globally smooth temperature approximation may therefore impose more regularity than the physical solution possesses. Separate subdomain fields with explicit transmission constraints can represent the correct jump structure. If contact resistance is present, a temperature jump can be physical, and enforcing continuity would be incorrect.

Analogous issues arise in solid mechanics. Displacement continuity and traction equilibrium across a bonded interface do not require identical strain or stress components on both sides. Mixed stress-displacement formulations introduce constitutive consistency in addition to equilibrium [16, 17]. Domain decomposition and finite-basis PINNs localize approximation, while variational and hp-variational approaches alter the residual testing procedure [18, 19, 20]. These methods provide different tools for respecting regularity; none should be selected solely because it reduces a displayed loss.

Arora et al.'s viscoplastic model illustrates the importance of internal variables, rate dependence, temperature, and balanced mechanical residuals [21]. A displacement field alone need not encode the complete constitutive history. Exact enforcement of a kinematic boundary condition does not ensure correct plastic evolution or a physically admissible strength variable. Initial internal states and evolution equations must remain part of the formulation and verification.

Thermodynamics-based neural constitutive models can derive stress and other quantities from structured potentials [22]. A potential representation can enforce selected derivative relationships,

but additional restrictions may be needed for convexity, positivity, objectivity, or nonnegative dissipation. One should identify which property follows mathematically from the architecture and which is only encouraged or tested. Checking a few loading paths is evidence over those paths, not a proof of global constitutive admissibility.

## **7. When boundary observations are uncertain**

Hard enforcement is most straightforward when the prescribed condition is a trusted modeling input. A noisy temperature sensor on a boundary is a different object. Forcing a neural field to reproduce every noisy reading exactly can introduce an irregular boundary trace and distort the interior solution. A likelihood or weighted observation loss can instead represent the measurement uncertainty. Another option is a latent smooth boundary field constrained by measurements and physical knowledge.

Bayesian PINNs provide one framework for representing uncertain observations and inferred quantities [23]. The boundary uncertainty model should still reflect systematic offsets, correlations, and unknown transfer coefficients when those are material. Exact enforcement of an incorrect nominal boundary can yield a confident but biased interior prediction. Inverse-problem stability analyses make clear that the observation arrangement and physical problem determine what can be recovered reliably [24].

The choice is therefore not simply exact versus approximate physics. One may impose exact conservation while treating uncertain boundary measurements statistically. A prescribed symmetry condition can be exact, while an estimated heat-transfer coefficient remains an inferred parameter. Documenting this separation makes the model interpretable and prevents measurement uncertainty from being disguised as a physical law.

## **8. Verification and fair comparison**

A fair comparison should hold the physical problem, observations, training budget, and engineering evaluation fixed while changing the enforcement mechanism. Compare a normalized soft penalty, an exact admissible ansatz where available, and a constrained optimization method where appropriate. Report boundary or constraint violation, interior residuals, engineering quantities of interest, and total computational cost. The exact ansatz wins the enforced-trace metric by construction; the substantive question is its effect on the remaining accuracy and cost.

Use independent verification points and suitable quadrature, especially near corners, interfaces, and small constrained boundary segments. Adaptive and importance-sampling approaches can improve allocation of training points, but cannot certify unsampled regions by themselves [25, 26]. A finite residual average should not be presented as a bound on a continuous constraint norm without additional assumptions. Linear-PDE convergence results provide qualified guidance rather than blanket certification [27].

For time-dependent cases, verify initial-state compatibility and error evolution. Causal training changes how temporal residuals are prioritized, while gradient-enhanced training adds derivative information [28, 29]. Either can interact with a hard ansatz, but these interventions should be

separated in ablation studies. Published training guidance helps standardize these choices [30]. A comparison that changes enforcement, optimizer, architecture, and time strategy together cannot identify the cause of an improvement.

Benchmark suites and reusable frameworks support reproducible implementation [31, 32, 33]. Conventional finite-element or finite-volume baselines remain essential, since PINNs do not generally dominate established solvers in forward-solution accuracy or time [34]. Report differentiation overhead and geometry-construction cost as well as training iterations. An exact ansatz that requires expensive geometry processing may still be worthwhile, but that cost belongs in the comparison.

| Engineering condition                          | Suitable enforcement candidate                         | Required independent check                            |
|------------------------------------------------|--------------------------------------------------------|-------------------------------------------------------|
| Trusted prescribed displacement or temperature | Boundary lifting or distance-based ansatz              | Derivatives, reactions, or heat flux                  |
| Incompressibility                              | Potential representation when appropriate              | Momentum, wall conditions, pressure reference         |
| Noisy boundary measurements                    | Statistical observation model or latent boundary field | Predictive coverage and sensitivity to bias           |
| Bonded material interface                      | Separate fields with transmission conditions           | Flux or traction balance and allowed derivative jumps |
| Constitutive evolution                         | Structured state variables and evolution constraints   | Loading-path response and admissibility               |

### 9. What exact boundary values contribute to an error argument

For a simple Poisson problem, exact homogeneous error boundary conditions allow a useful energy identity. Let  $u$  solve  $-\Delta u = f$  with a prescribed Dirichlet trace, let  $u_h$  have that same trace, and define  $e = u_h - u$  and  $r = -\Delta u_h - f$ . Assuming the regularity needed for integration by parts, the error satisfies

$$\int_{\Omega} |\nabla e|^2 \, dx = \int_{\Omega} r \, e \, dx.$$

$$\| \nabla e \|_{L^2} \leq C_P \| r \|_{L^2}.$$

The second line uses the Cauchy–Schwarz inequality and the Poincaré inequality  $\|e\| \leq C_P \| \nabla e \|$  for the stated zero-trace space. If the gradient error is zero, the bound is immediate; otherwise division by its norm gives the result. This supplies a concrete benefit of exact Dirichlet enforcement: no unknown boundary-error term appears in this particular argument. It still requires a continuous residual norm and a domain-dependent constant, and it does not turn a small collocation average into a certified bound.

If the trace is only approximate, an error lifting or a boundary term must enter the analysis. The relevant boundary norm depends on the PDE and desired solution norm. A pointwise boundary MSE may not control it without further regularity and sampling assumptions. Therefore, comparing hard and soft models solely through their interior residuals can be misleading; the soft model carries an additional source of error that the hard model removes by construction.

For a practical conduction study, use a known heat source, a trusted prescribed-temperature boundary, and an independently resolved reference solution. Compare the three enforcement mechanisms under the same computational budget. Verify the boundary trace, the domain energy balance, the temperature field, and the boundary heat flux. Then repeat with deliberately uncertain boundary measurements using a documented noise model. The exact prescribed boundary and the uncertain measured boundary are different inverse or forward problems and should not be combined into one performance ranking.

This proposed study would show both the benefit and the limits of exactness. The exact ansatz should satisfy its prescribed trace to numerical precision, but its heat-flux accuracy and training cost remain empirical questions. The noisy-boundary case tests whether treating measurements statistically improves prediction when the imposed values are imperfect. Such a design makes the engineering implications observable without assuming in advance that one enforcement mechanism must win.

## 10. Conclusions

Exact parameterizations, penalties, and constrained optimization enforce different mathematical objects. Exact boundary lifting can remove a trace error while changing derivative conditioning. Finite penalties trade feasibility against optimization difficulty. Compatibility and gauge conditions determine whether the physical problem has a solution before any learning begins.

For engineering applications, constraints should be selected according to trusted information, physical regularity, and the intended quantity of interest. A rigorous study states what is exact by construction, measures what is only approximately enforced, and verifies the remaining equations independently. That distinction is more useful than treating hard constraints as universally superior to soft ones.

## References

- [1] Maziar Raissi, Paris Perdikaris, George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378, 686-707 (2019). DOI.
- [2] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, et al.. Physics-informed machine learning. *Nature Reviews Physics* 3, 422-440 (2021). DOI.
- [3] N. Sukumar, Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. arXiv:2104.08426 (2021). arXiv.
- [4] Lu Lu, Raphael Pestourie, Wenjie Yao, et al.. Physics-informed neural networks with hard constraints for inverse design. arXiv:2102.04626 (2021). arXiv.

- [5] Shuheng Liao, Tianju Xue, Jihoon Jeong, et al.. Hybrid thermal modeling of additive manufacturing processes using physics-informed neural networks for temperature prediction and parameter identification. arXiv:2206.07756 (2022). arXiv.
- [6] Qiming Zhu, Zeliang Liu, Jinhui Yan. Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks. arXiv:2008.13547 (2020). arXiv.
- [7] Sifan Wang, Yujun Teng, Paris Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv:2001.04536 (2020). arXiv.
- [8] Sifan Wang, Xinling Yu, Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. arXiv:2007.14527 (2020). arXiv.
- [9] Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, et al.. Characterizing possible failure modes in physics-informed neural networks. arXiv:2109.01050 (2021). arXiv.
- [10] Levi McClenny, Ulisses Braga-Neto. Self-Adaptive Physics-Informed Neural Networks using a Soft Attention Mechanism. arXiv:2009.04544 (2020). arXiv.
- [11] Rafael Bischof, Michael Kraus. Multi-Objective Loss Balancing for Physics-Informed Deep Learning. arXiv:2110.09813 (2021). arXiv.
- [12] Chengping Rao, Hao Sun, Yang Liu. Physics-informed deep learning for incompressible laminar flows. arXiv:2002.10558 (2020). arXiv.
- [13] Maziar Raissi, Alireza Yazdani, George Em Karniadakis. Hidden Fluid Mechanics: A Navier-Stokes Informed Deep Learning Framework for Assimilating Flow Visualization Data. arXiv:1808.04327 (2018). arXiv.
- [14] Maziar Raissi, Zhicheng Wang, Michael S. Triantafyllou, George Em Karniadakis. Deep Learning of Vortex Induced Vibrations. arXiv:1808.08952 (2018). arXiv.
- [15] Shengze Cai, Zhiping Mao, Zhicheng Wang, et al.. Physics-informed neural networks (PINNs) for fluid mechanics: A review. arXiv:2105.09506 (2021). arXiv.
- [16] Ehsan Haghighat, Maziar Raissi, Adrian Moure, et al.. A deep learning framework for solution and discovery in solid mechanics. arXiv:2003.02751 (2020). arXiv.
- [17] R. Sharma, Y. B. Guo. Thermal-Mechanical Physics Informed Deep Learning For Fast Prediction of Thermal Stress Evolution in Laser Metal Deposition. arXiv:2412.18786 (2024). arXiv.
- [18] Ben Moseley, Andrew Markham, Tarje Nissen-Meyer. Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. arXiv:2107.07871 (2021). arXiv.
- [19] E. Kharazmi, Z. Zhang, G. E. Karniadakis. Variational Physics-Informed Neural Networks For Solving Partial Differential Equations. arXiv:1912.00873 (2019). arXiv.
- [20] Ehsan Kharazmi, Zhongqiang Zhang, George Em Karniadakis. hp-VPINNs: Variational Physics-Informed Neural Networks With Domain Decomposition. arXiv:2003.05385 (2020). arXiv.
- [21] Rajat Arora, Pratik Kakkar, Amit Chakraborty, Biswadip Dey. Physics-informed neural networks for modeling rate- and temperature-dependent plasticity. Machine Learning and the Physical Sciences Workshop, NeurIPS 2022. arXiv:2201.08363v3 (2022). arXiv.
- [22] Filippo Masi, Ioannis Stefanou, Paolo Vannucci, Victor Maffi-Berthier. Thermodynamics-based Artificial Neural Networks for constitutive modeling. arXiv:2005.12183 (2020). arXiv.
- [23] Liu Yang, Xuhui Meng, George Em Karniadakis. B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data. arXiv:2003.06097 (2020). arXiv.
- [24] Siddhartha Mishra, Roberto Molinaro. Estimates on the generalization error of Physics Informed Neural Networks (PINNs) for approximating a class of inverse problems for PDEs. arXiv:2007.01138 (2020). arXiv.
- [25] Chenxi Wu, Min Zhu, Qinyang Tan, et al.. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. arXiv:2207.10289 (2022). arXiv.

- [26] Mohammad Amin Nabian, Rini Jasmine Gladstone, Hadi Meidani. Efficient training of physics-informed neural networks via importance sampling. arXiv:2104.12325 (2021). arXiv.
- [27] Yeonjong Shin, Jerome Darbon, George Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. arXiv:2004.01806 (2020). arXiv.
- [28] Sifan Wang, Shyam Sankaran, Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. arXiv:2203.07404 (2022). arXiv.
- [29] Jeremy Yu, Lu Lu, Xuhui Meng, George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. arXiv:2111.02801 (2021). arXiv.
- [30] Sifan Wang, Shyam Sankaran, Hanwen Wang, Paris Perdikaris. An Expert's Guide to Training Physics-informed Neural Networks. arXiv:2308.08468 (2023). arXiv.
- [31] Zhongkai Hao, Jiachen Yao, Chang Su, et al.. PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs. arXiv:2306.08827 (2023). arXiv.
- [32] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, et al.. PDEBENCH: An Extensive Benchmark for Scientific Machine Learning. arXiv:2210.07182 (2022). arXiv.
- [33] Lu Lu, Xuhui Meng, Zhiping Mao, George E. Karniadakis. DeepXDE: A deep learning library for solving differential equations. arXiv:1907.04502 (2019). arXiv.
- [34] Tamara G. Grossmann, Urszula Julia Komorowska, Jonas Latz, Carola-Bibiane Schönlieb. Can Physics-Informed Neural Networks beat the Finite Element Method?. arXiv:2302.04107 (2023). arXiv.
- [35] Tongtao Zhang, Biswadip Dey, Pratik Kakkar, Arindam Dasgupta, Amit Chakraborty. Frequency-compensated PINNs for Fluid-dynamic Design Problems. ML4Eng Workshop, NeurIPS 2020. arXiv:2011.01456 (2020). arXiv.