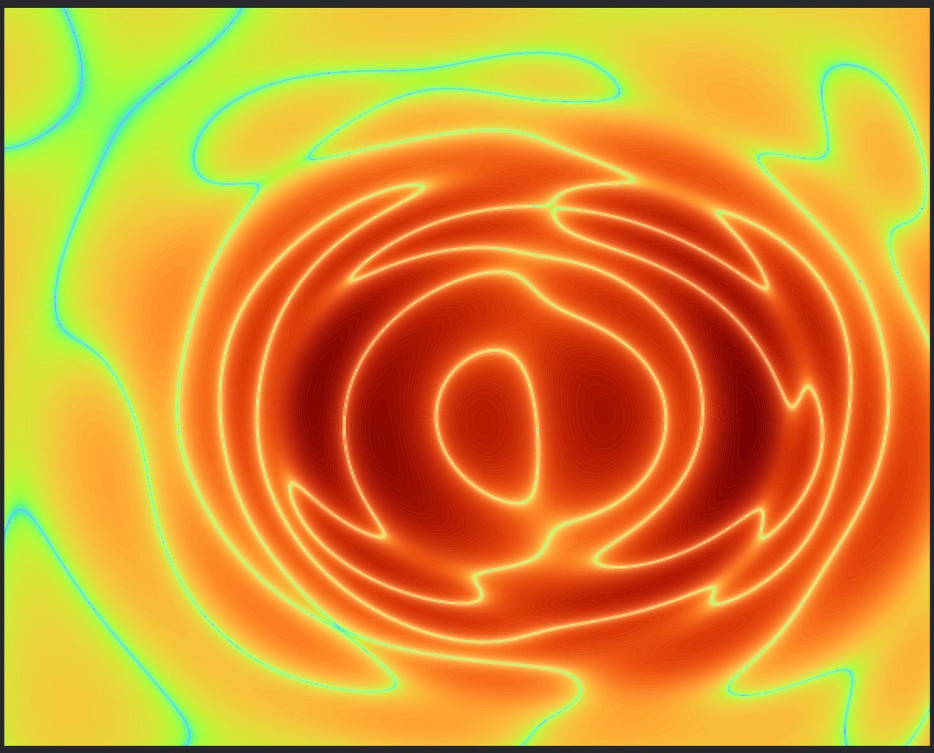

The divergence-form residual $\log_{10}|\mathrm{div}(a\,\mathrm{grad}\,u)+f|$ of a verified deep-learning solver — a physics-informed network's own error blooming across a five-dimensional high-contrast inclusion, exactly where Monte Carlo pins it down.

Verified Deep Learning Solvers for PDEs

PINNs in PyTorch and DeepXDE, Checked by Monte Carlo

Amnon Gershon

ODIN PRESS

Verified Deep Learning Solvers for PDEs

Paperback version, First edition: 2026.
ISBN: forthcoming (Bowker allocation pending)
Trim Size: 6×9 inch. Paper color: White.

Published by Odin Press.
`https://odinpress.org`
This book was typeset in LaTeX by Odin Press from a manuscript submitted by the author.

Printed and Bound in the United States of America.

9 8 7 6 5 4 3 2 1 0

Preface

A partial differential equation in three variables is solved on a grid. A partial differential equation in fifty variables is not — there is no grid, because the number of points one would need exceeds the number of atoms one has. This is the curse of dimensionality stated without ornament, and for most of my working life it marked the boundary of what deterministic numerical analysis could do. Inside the boundary we had finite differences and finite elements, and they were excellent. Outside it we had almost nothing, and we knew it.

Neural networks crossed that boundary. A physics-informed network represents the solution of a PDE not on a mesh but as a function, parameterized by weights and trained to make the equation's residual small at scattered collocation points. Because the representation is mesh-free, the dimension that defeats a grid no longer defeats the solver in the same way. Networks now produce solutions to high-dimensional semilinear equations, to committor problems, to Schrödinger ground states, that no grid I ever built could have reached. The advance is real. I do not dispute it, and this book makes daily use of it.

But a trained network comes with no error certificate. This is the gap the book exists to close. A finite-element solution carries a posteriori error estimates; a physics-informed network carries a training loss, and a training loss is not an error. I have watched a network drive its residual to near-zero on its collocation points while its solution was visibly wrong between them — in a domain those points could never have covered. A wrong solution and a right one look identical until something independent is brought to bear. Early in my career a finite-difference scheme converged cleanly, smoothly, confidently, to the wrong answer, and nothing in the residual warned me; a crude Monte Carlo estimate did. I have not trusted a single solver's word for anything since.

The independent check, in this book, is the one probability theory pro-

vides for free. A large class of partial differential equations — the linear and semilinear elliptic and parabolic equations, the eigenvalue problems, the stationary densities — have a stochastic representation: their solution at a point equals an expectation over the paths of a diffusion, through the Feynman–Kac correspondence. That expectation can be estimated by simulating the diffusion and averaging, and the estimate comes with a standard error and a confidence interval attached. So for every PDE that has such a twin, we have two unrelated solvers: the network, and the Monte Carlo estimator. The thesis of this book is that you put their answers, with the Monte Carlo confidence interval, on the same page — and you believe the network only where the two agree. A solution two unrelated methods agree on is a solution; a solution standing on one method alone is a conjecture.

What I found, working through the problems collected here, is a fact I had half-expected and now state plainly: the partial differential equations that deep learning solves well are very nearly the same equations that Monte Carlo can check. Both methods love structure — a stochastic representation, a variational principle, a diffusion behind the operator. The overlap is not a coincidence. It is the whole story, and it is the reason the verification programme is feasible rather than aspirational. Where the overlap holds, the book reports agreement to within the stated standard error, problem after problem. Where it fails — advection-dominated transport, sharp fronts, shocks, the incompressible pressure — neither the network nor the check is trustworthy, and the final chapter says so without softening it. An honest book about a method must include the cases where the method breaks.

The chapters proceed from the verifiable to the unverifiable. Chapter 1 builds the bridge: the Feynman–Kac correspondence and the Monte Carlo twin that every later chapter leans on. Chapters 2 and 3 take the simplest elliptic problems — steady heat conduction and electrostatics in awkward geometry — and check Poisson and Laplace solutions against walk-on-spheres simulation. Chapter 4 adds absorption and the killed diffusion that represents it; Chapter 5 turns the exit time itself into the quantity of interest, the mean first-passage time out of a potential well. Chapter 6 reaches the committor function and the rare-event simulation behind transition paths. Chapter 7 lets the coefficients vary in space; Chapter 8 lets the equation become nonlinear, the Fisher–KPP front and its branching-Brownian-motion twin. Chapter 9 is the eigenvalue problem — the Schrödinger ground state, checked by diffusion Monte Carlo, the method that once caught a spectral solver settling quietly on an excited state. Chapters 10 through 13 are the

high-dimensional payoff: the deep BSDE method for semilinear PDEs, the network used as a control variate to accelerate the Monte Carlo rather than merely to be checked by it, a basket option in dimensions no grid admits, and the stationary Fokker–Planck density of a long-run equilibrium. Chapter 14 is where it breaks, and it is the chapter I would ask a sceptical reader to read first.

Every number in this book was produced by code, and every figure is reproducible from a single self-contained file under a fixed seed — SEED = 1000 throughout — or it is an anecdote and does not appear. The decisive script for each chapter is printed in the appendix; the full simulators live in the book's code repository. I am not impressed by a low training loss. I am impressed by an independent check that passes, and I have tried to give you nothing in these pages that I have not checked.

— Amnon Gershon, Professor Emeritus

Contents

Chapter 1

The Bridge: PDEs with a Monte Carlo Twin

A physics-informed neural network returns a field, a training history, and a final loss of perhaps 10^{-5}. It does not return an error. The loss records how nearly the network satisfied its equation at the finite set of collocation points where it was asked; it is silent about the points in between, and silent altogether about the distance from the field it produced to the field one wanted. Those are different quantities, related — when they are related at all — by a stability constant of the operator that the training never measures. A residual driven to 10^{-5} is therefore consistent with a solution that is everywhere accurate and with one that is visibly wrong, and nothing on the training curve tells the two apart.

A solver one cannot independently check has produced a conjecture, not a solution, however small its residual. That conviction is the premise of this book, and for one large and important class of equations it comes with a constructive remedy. For the diffusive equations — those whose solution is an average of its data over a randomly spreading path — probability theory supplies a second solver that shares none of the network's machinery: the Monte Carlo estimate of that average, which by its nature arrives with a confidence interval. The two methods have no common construction and no common bias, so a network value that lands inside the interval has been checked against an independent witness, and one that lands outside has been caught. The interval is the a posteriori error certificate the residual could not supply.

The remedy has an exact boundary, and the book is organized around it

rather than around concealing it. The diffusive equations — heat, Poisson, Fokker–Planck, and the high-dimensional semilinear and pricing equations of the later chapters — are precisely the ones carrying a stochastic representation, and precisely the ones a Monte Carlo estimate can check; that the two classes coincide is not luck but the structure of the subject. The equations with no diffusion to represent them — pure advection, sharp fronts, incompressible pressure — have no such twin, and where the check is absent the network is to be trusted less, not more. Chapter 14 maps that boundary in full; the present chapter stays well inside it.

The six sections build the test in order. Section 1.1 states the gap precisely — why a small residual carries no bound on the solution error. Section 1.2 introduces the Feynman–Kac representation that rewrites a parabolic equation as an expectation, and reads off which equations carry a twin and which, lacking a diffusion, do not. Section 1.3 trains the network in DeepXDE; Section 1.4 constructs the Monte Carlo estimate and its interval; Section 1.5 joins the two into a single acceptance rule and exhibits it on a solver that passes and, as plainly, on one that fails. Section 1.6 fixes the reproducibility contract — a single seed `SEED` $= 1000$, isolated random streams, deterministic execution — and explains why the experiments default to five dimensions. Five is high enough that a tensor grid is already hopeless, so the mesh-free network and its Monte Carlo check are doing real work, and low enough that an exact solution still exists to hold both against. That exact reference is a luxury of this chapter alone; from Chapter 2 onward it is gone, and the confidence interval is the only ground truth there is.

1.1 The Problem with a Small Residual: A Solver Without an Error Certificate

Let L be a differential operator, $\Omega \subset \mathbb{R}^d$ a domain carrying boundary or initial data, and u the field we want. The object of study throughout the book is the forward problem in the following precise sense.

Definition 1.1 (Forward PDE problem)**.** A *forward PDE problem* is a triple (L, Ω, data): a differential operator L, a domain Ω with prescribed boundary and/or initial data, and the unknown field u required to satisfy $Lu = f$ on Ω together with the prescribed conditions on $\partial\Omega$. The field u is the solution; f is the source.

A physics-informed neural network attacks this problem by turning the operator into a loss. It replaces u with a network and asks that the network make $Lu - f$ small where it is sampled.

Definition 1.2 (Physics-informed neural network)**.** A *physics-informed neural network* (PINN) is a parametric map $u_\theta : \Omega \to \mathbb{R}$, typically a fully connected network, trained by minimizing the *residual loss*

$$\mathcal{L}(\theta) \;=\; \frac{1}{N_r}\sum_{i=1}^{N_r}\big(Lu_\theta(x_i) - f(x_i)\big)^2 \;+\; \lambda\,\frac{1}{N_b}\sum_{j=1}^{N_b}\big(u_\theta(y_j) - g(y_j)\big)^2, \quad (1.1)$$

where the x_i are *collocation points* sampled in Ω, the y_j lie on $\partial\Omega$ with boundary data g, and $\lambda > 0$ weights the boundary term [1, 2]. The idea of solving differential equations by training a neural network to satisfy them dates to [3] and is surveyed in [4, 5].

The training minimizes $\mathcal{L}(\theta)$ and reports its final value. The temptation — and it is the error this whole book exists to correct — is to read a small final loss as a small solution error. The two are different quantities measured in different norms, and nothing in the optimization relates them.

Proposition 1.3 (No a posteriori error certificate)**.** A small residual loss at finitely many collocation points does not bound the global solution error. There is no constant C, available from the training, for which $\|u_\theta - u\| \le C\,\mathcal{L}(\theta)^{1/2}$ holds uniformly; the network carries no a posteriori error certificate.

Proof. Two obstructions, either one sufficient. First, the loss is a sum over a finite point set $\{x_i\}$; the residual $Lu_\theta - f$ is constrained only there and is free to take any values between the points. The map from the pointwise residual sample to the global error $\|u_\theta - u\|$ is therefore not even well defined without further regularity assumptions on u_θ, and a network with bounded loss can carry arbitrarily large residual off the sample. Second, even with the residual controlled in a continuous norm, bounding $\|u_\theta - u\|$ by $\|Lu_\theta - f\|$ requires a stability estimate for L^{-1} — a constant $C = \|L^{-1}\|$ in the appropriate norms. That constant is a property of the operator and the domain, not of the network, and the training never estimates it. For an operator with a large or unknown stability constant — a convection-dominated problem, a thin boundary layer — a residual driven to 10^{-5} is consistent with an order-one error. The training measures $\mathcal{L}(\theta)$; it does not measure $\|u_\theta - u\|$, and it provides nothing to convert one into the other. □

Chapter 3

Electrostatics in Awkward Geometry: Laplace, Corners, and Mesh-Free Solving

The electrostatic potential between conductors is harmonic, so everything Chapter 2 built for steady heat carries over without a change of symbol: the same Laplace operator, the same Brownian exit representation, the same walk-on-spheres estimator. What changes is the boundary. A real enclosure is not all conductor. Its conducting electrodes fix the potential — a Dirichlet condition — but its insulating walls fix instead the normal component of the field to zero, a Neumann condition, and walk on spheres, which knows only how to be absorbed at a boundary, has nothing to do at a wall it is meant to bounce off.

This chapter delivers a verifier for that mixed problem. The probabilistic object behind a Neumann boundary is not absorbed Brownian motion but reflected Brownian motion, and the grid-free estimator that samples it is walk on stars, the recent extension of walk on spheres that handles insulating boundaries with the same closest-point machinery. We solve the electrostatics of a charged electrode in an awkward enclosure with a physics-informed network, and we check it two ways: against walk on spheres where the boundary is all conductor, and against walk on stars where an insulating wall reflects the path. Then we push past the potential to the field and

the capacitance, where a derivative of the solution is the stricter test. Section 3.1 casts electrostatics as the harmonic problem of Chapter 2. Section 3.2 introduces the mixed Dirichlet–Neumann boundary and why the insulating wall needs new machinery. Section 3.3 builds walk on stars and its reflecting-Brownian-motion interpretation. Section 3.4 trains the mixed-condition network and names its soft-Neumann weak spot. Section 3.5 runs the verification near the walls. Section 3.6 recovers the field and the capacitance and tests them.

3.1 Electrostatics as an Elliptic Problem: The Harmonic Potential

Let $\Omega \subset \mathbb{R}^2$ be the charge-free region between conductors, $u(x)$ the electrostatic potential, and $E = -\nabla u$ the electric field. Gauss's law and the definition of the potential together fix the equation u must satisfy, and in the charge-free case it is the one we already know how to verify.

Definition 3.1 (Electrostatic potential)**.** The *electrostatic potential* u in a region with charge density ρ solves Poisson's equation $-\Delta u = \rho/\varepsilon_0$, and the field is $E = -\nabla u$. In a charge-free region $\rho \equiv 0$ it solves *Laplace's equation* $\Delta u = 0$ and the potential is harmonic [14, 20].

Proof. Gauss's law in differential form is $\nabla \cdot E = \rho/\varepsilon_0$. Substituting $E = -\nabla u$ gives $\nabla \cdot (-\nabla u) = -\Delta u = \rho/\varepsilon_0$, which is Poisson's equation; with $\rho \equiv 0$ the right-hand side vanishes and the potential satisfies $\Delta u = 0$. This is the citation's statement of electrostatics as a Laplace/Poisson problem for the potential [14]. □

The consequence is immediate and worth stating, because it means the entire apparatus of the previous chapter transfers with no probabilistic rework.

Proposition 3.2 (The conductor problem is Chapter 2's problem)**.** Between conductors the potential is harmonic, so Kakutani's exit representation $u(x) = \mathbb{E}[g(B_\tau)]$ and the walk-on-spheres estimator of Chapter 2 apply to it unchanged: the potential at an interior point is the expected conductor potential at the Brownian exit point.

Proof. By the definition above, in a charge-free region u is harmonic. Chapter 2's Kakutani theorem holds for any harmonic function on a bounded

domain with continuous Dirichlet data: the proof used only $\Delta u = 0$ and the martingale property of $u(B_t)$ along Brownian motion, neither of which refers to heat. Taking the conductor potentials as the boundary data g, the representation $u(x) = \mathbb{E}[g(B_\tau)]$ and its walk-on-spheres estimator transfer verbatim. The physical reading changes — boundary temperature becomes conductor voltage — but the mathematics is identical. □

The maximum principle transfers with it, and in the electrostatic setting it is a statement every physicist already trusts.

Proposition 3.3 (No interior voltage extremum)**.** In a charge-free region the potential attains no local maximum or minimum at an interior point: the highest and lowest voltages live on the conductors. An interior value outside the range of the conductor potentials is therefore unphysical.

Proof. The potential is harmonic, so by the maximum principle of Chapter 2 it attains its extrema on $\partial\Omega$; equivalently, $u(x) = \mathbb{E}[g(B_\tau)]$ is an average of conductor potentials and lies between their minimum and maximum. A local interior maximum would be a charge-free point where field lines converge from all directions, which Gauss's law forbids without a charge to terminate them. Any solver returning an interior voltage above the hottest electrode or below the coldest has produced an unphysical field. □

Corollary 3.4 (The potential is bracketed by the electrode voltages)**.** For the running enclosure, with the electrode at $u = 1$ and the grounded walls at $u = 0$, every interior potential satisfies $0 \le u(x) \le 1$.

Proof. The boundary data takes the two values 0 and 1, so $\min_{\partial\Omega} g = 0$ and $\max_{\partial\Omega} g = 1$; the bracketing proposition gives $0 \le u(x) \le 1$ at every interior point. Any network or Monte Carlo value outside $[0, 1]$ is rejected before any interval is computed. □

The geometry we run this on is chosen to be awkward in a way conductors really are.

Example 3.5 (A charged electrode in a grounded enclosure)**.** The running geometry is the box $\Omega = [-1, 1]^2$ with a square electrode $[-0.25, 0.25]^2$ removed from its center and held at $u = 1$; the top and bottom walls $y = \pm 1$ are grounded at $u = 0$. The electrode's four corners are sharp, and at a sharp conductor the field concentrates — the classic lightning-rod effect — so the geometry has built into it exactly the regions where a smooth network will struggle, and where the verifier must be trusted to find the struggle.

A trained neural network can solve a partial differential equation in fifty dimensions, where no grid can be built. But it hands you no error certificate. Its training loss is not its error, and a wrong solution looks exactly like a right one until something independent checks it.

This book supplies the check. For every PDE with a stochastic representation — the elliptic and parabolic equations, the eigenvalue problems, the stationary densities — the Feynman–Kac correspondence gives a Monte Carlo twin whose estimate comes with a confidence interval for free. The method is simple and strict: solve with the network, estimate with Monte Carlo, put both on the same page, and believe the network only where the two agree within the standard error.

Across fourteen worked problems — from steady heat conduction to the deep BSDE method, basket-option pricing, and Schrödinger ground states — every solution is verified or declared unverifiable. The final chapter is candid about where the whole programme breaks: advection, shocks, sharp fronts. Every figure is reproducible from one self-contained file under a fixed seed.

For engineers, quants, and computational scientists who refuse to trust a solver they cannot check.

— Amnon Gershon, Professor Emeritus