40 Models for Mapping Inputs → Outputs

A compact field-guide collecting **formulations**, **where they're used**, **history**, **use cases**, and **what they comprise**—all in one place.

Notation (lightweight): inputs i (or sequence $i_{1:T}$), outputs o (or $o_{1:T}$); dataset $\mathcal{D} = \{(i_k, o_k)\}$. Vectors are bold implicitly; noise terms ϵ, w, v as context requires.

---

1) Deterministic function (static map)

Formulation. $o_k = f(i_k)$. - **Where:** rule systems, ETL transforms, calculators.
- **History:** classical baseline from mathematical analysis.
- **Use cases:** scoring rules, lookup tables, calibrations.
- **Comprises:** explicit rule set or closed-form expression; no estimation.

2) Linear / affine model

Formulation. $o_k = W i_k + b$ (vector) or $o_k = w^\top i_k + b$ (scalar). - **Where:** baselines across science/engineering.
- **History:** Gauss/Legendre, early 1800s (least squares).
- **Use cases:** prediction, forecasting, control.
- **Comprises:** weights, bias; least squares or regularized fitting.

3) Basis expansion (poly/splines/Fourier)

Formulation. $o_k = \sum_j \alpha_j \, \phi_j(i_k)$. - **Where:** statistics, signal processing, curve fitting.
- **History:** Fourier (19th c.); splines (Schoenberg, 1946).
- **Use cases:** nonlinearity with interpretability, seasonality, smooth trends.
- **Comprises:** dictionary of basis functions; linear coefficients; regularization.

4) k-Nearest Neighbors (kNN)

Formulation. $\hat{o}(i) = \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} o_j$. - **Where:** low-dimensional data; quick baselines.
- **History:** Fix & Hodges (1951).
- **Use cases:** classification/regression, imputation.
- **Comprises:** distance metric, k, tie-breaking; no training phase beyond indexing.

5) Kernel regression (Nadaraya–Watson)

Formulation. $\hat{o}(i) = \dfrac{\sum_j K(i, i_j) \, o_j}{\sum_j K(i, i_j)}$. - **Where:** smoothing/LOESS-style fits.
- **History:** 1964.

- **Use cases:** calibration curves, local averaging.
- **Comprises:** kernel choice and bandwidth; sometimes local polynomial variants.

6) Logistic / softmax models

Formulation. Binary: $\Pr(o = 1 \mid i) = \sigma(w^\top i + b)$. Multiclass: $\Pr(o = c \mid i) = \frac{e^{w_c^\top i + b_c}}{\sum_{c'} e^{w_{c'}^\top i + b_{c'}}}$. - **Where:** classification, propensity modeling.
- **History:** logistic curve (19th c.), logistic regression (1950s).
- **Use cases:** credit risk, medical diagnosis, click-through.
- **Comprises:** linear scores + link; cross-entropy; regularization.

7) Decision trees / ensembles (RF/GBDT)

Formulation. Piecewise model $o = \sum_\ell \theta_\ell \, \mathbf{1}[i \in R_\ell]$; ensembles average/vote.
- **Where:** tabular data.
- **History:** CART (1984); Random Forests (2001); Gradient Boosting (2001+).
- **Use cases:** ranking, churn, credit scoring, feature importance.
- **Comprises:** splits, impurity metrics; bagging/boosting, shrinkage, depth/trees.

8) Gaussian processes (GP)

Formulation. $f \sim \mathcal{GP}(m, k)$, $o = f(i) + \epsilon$. Predictive mean $\mu_* = k_*^\top (K + \sigma^2 I)^{-1} y$; variance $\sigma_*^2 = k(i_*, i_*) - k_*^\top (K + \sigma^2 I)^{-1} k_*$. - **Where:** small/medium data with uncertainty needs.
- **History:** Kriging (1930s); modern GPs (1990s–2000s).
- **Use cases:** Bayesian optimization, spatial stats, calibration.
- **Comprises:** kernel, mean function; exact or sparse inference.

9) Mixture of experts (MoE)

Formulation. $p(o \mid i) = \sum_{m=1}^{M} \pi_m(i) \, p_m(o \mid i)$. - **Where:** heterogeneous regimes, large models (routing).
- **History:** Jacobs/Jordan (1991); modern sparse MoE for scaling.
- **Use cases:** multimodal outputs, domain specialization.
- **Comprises:** experts (NNs/GLMs), gating network, mixture training.

10) Bayesian likelihood model

Formulation. Prior $p(\theta)$, likelihood $p(o \mid i, \theta)$; posterior $p(\theta \mid \mathcal{D}) \propto p(\theta) \prod_k p(o_k \mid i_k, \theta)$. - **Where:** principled uncertainty, hierarchical pooling.
- **History:** Bayes (1763); MCMC/VI (1990s+).
- **Use cases:** A/B tests, meta-analysis, small-data regimes.
- **Comprises:** prior, likelihood family, inference (MCMC/VI), posterior predictive.

11) Noisy memoryless channel

Formulation. $p(o \mid i)$; analyze capacity $\max_{p(i)} I(I;O)$. - **Where:** communications, error modeling (OCR/ASR).
- **History:** Shannon (1948).
- **Use cases:** coding, denoising, robust decoding.
- **Comprises:** channel law (e.g., BSC, AWGN), information measures.

12) Causal SEM / DAGs

Formulation. For variables X_j : $X_j = f_j(\mathrm{Pa}(X_j), U_j)$, acyclic graph; interventions via $do(\cdot)$. - **Where:** policy, epidemiology, economics.
- **History:** Wright (1920s), Pearl (1990s).
- **Use cases:** counterfactuals, mediation, uplift.
- **Comprises:** graph, structural equations, exogenous noise, identification.

13) LTI / convolutional system

Formulation. $o_t = (h * i)_t = \sum_{\tau \geq 0} h_\tau \, i_{t-\tau}$; transfer $H(z) = \sum_\tau h_\tau z^{-\tau}$. - **Where:** control, DSP, audio, imaging.
- **History:** 1930s–1950s systems theory.
- **Use cases:** filters, deconvolution, equalization.
- **Comprises:** impulse response, stability, frequency response.

14) Linear state-space (Kalman)

Formulation. $x_{t+1} = Ax_t + Bi_t + w_t$; $o_t = Cx_t + Di_t + v_t$.
- **Where:** tracking, navigation, sensor fusion.
- **History:** Kalman (1960).
- **Use cases:** robotics, aerospace, econometrics nowcasting.
- **Comprises:** system matrices, noise covariances; Kalman filter/smoother.

15) ARIMAX / dynamic regression

Formulation. $o_t = \sum_{j=1}^{p} \phi_j o_{t-j} + \sum_{m=0}^{q} \beta_m i_{t-m} + \epsilon_t$ (with differencing/MA terms as needed). - **Where:** forecasting with exogenous drivers.
- **History:** Box–Jenkins (1970).
- **Use cases:** demand, price, traffic forecasting.
- **Comprises:** AR/MA orders, differencing, exogenous regressors.

16) Input–Output HMM (IO-HMM)

Formulation. $p(z_{t+1} \mid z_t, i_t)$, $p(o_t \mid z_t, i_t)$; latent z_t . - **Where:** controlled/semi-Markov sequences.
- **History:** 1990s extensions of HMMs.

- **Use cases:** dialogue systems, bio/finance regimes.
- **Comprises:** state set, input-conditioned transitions/emissions; EM/inference.

17) RNN / LSTM / GRU

Formulation. $h_t = \phi(W_{ih} i_t + W_{hh} h_{t-1} + b)$, $o_t = W_{ho} h_t + c$. LSTM/GRU add gates.
- **Where:** sequential ML.
- **History:** RNN (1980s), LSTM (1997), GRU (2014).
- **Use cases:** language, sensor streams, anomaly detection.
- **Comprises:** recurrent cell, hidden state, optimizer, regularization.

18) Transformer (seq2seq/attention)

Formulation. $\mathrm{Attn}(Q, K, V) = \mathrm{softmax}(QK^\top/\sqrt{d})V$; encoder/decoder stacks with positional encodings.
- **Where:** NLP, vision, time-series.
- **History:** 2017 onward.
- **Use cases:** translation, summarization, forecasting, retrieval.
- **Comprises:** self/cross-attention blocks, MLPs, normalization, residuals.

19) Contextual bandit

Formulation. Choose action $a_t \in \mathcal{A}$ from context x_t ; observe reward $r_t \sim p(r \mid x_t, a_t)$; learn policy $\pi(a \mid x)$. - **Where:** online decisioning.
- **History:** 2000s; LinUCB/Thompson sampling variants.
- **Use cases:** recommendations, ads, UI optimization.
- **Comprises:** exploration (UCB/TS), reward model, regret analysis.

20) MDP / Reinforcement learning

Formulation. $p(s_{t+1} \mid s_t, a_t)$; objective $\max_\pi \mathbb{E}[\sum_t \gamma^t r_t]$. - **Where:** control, operations, games.
- **History:** Bellman (1950s); Sutton & Barto (1998).
- **Use cases:** robotics, inventory, scheduling, games.
- **Comprises:** state, action, reward, dynamics; value functions/policies.

21) Generalized linear model (GLM)

Formulation. $g(\mathbb{E}[o \mid i]) = \beta_0 + \beta^\top i$ with exponential-family likelihood. - **Where:** classical stats/actuarial.
- **History:** Nelder & Wedderburn (1972).
- **Use cases:** counts (Poisson), rates, insurance pricing.
- **Comprises:** link function, linear predictor, dispersion; MLE/IRLS.

22) Generalized additive model (GAM)

Formulation. $g(\mathbb{E}[o \mid i]) = \alpha + \sum_j s_j(i_j)$ with smoothers. - **Where:** interpretable nonlinear modeling.
- **History:** Hastie & Tibshirani (1986).
- **Use cases:** risk scores, uplift, partial dependence.
- **Comprises:** spline bases, penalties, backfitting.

23) Quantile regression

Formulation. $\min_\beta \sum_k \rho_\tau(o_k - \beta^\top i_k)$, $\rho_\tau(u) = \max\{\tau u, (\tau - 1)u\}$. - **Where:** distributional/risk estimates.
- **History:** Koenker & Bassett (1978).
- **Use cases:** VaR, service levels, asymmetric costs.
- **Comprises:** pinball loss, per-quantile fits or monotone joint fits.

24) SVM / SVR

Formulation (classification). $\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \sum \xi_k$ s.t. $y_k(w^\top \phi(i_k) + b) \geq 1 - \xi_k$.
Formulation (regression). ϵ -insensitive SVR with slack.
- **Where:** medium-size tabular/text.
- **History:** Vapnik (1990s).
- **Use cases:** margin-robust classification, robust regression.
- **Comprises:** kernel, support vectors, C/ϵ hyperparams.

25) MARS

Formulation. $o \approx \sum_m c_m B_m(i)$ where B_m are data-selected hinge bases.
- **Where:** response-surface modeling.
- **History:** Friedman (1991).
- **Use cases:** nonlinear tabular prediction with interpretability.
- **Comprises:** forward add / backward prune; GCV.

26) Isotonic regression

Formulation. Monotone f minimizing $\sum_k (o_k - f(i_k))^2$. - **Where:** monotone relationships, calibration.
- **History:** 1950s; pool-adjacent-violators algorithm (PAVA).
- **Use cases:** dose–response, score calibration.
- **Comprises:** monotonicity constraints; piecewise constant/linear fit.

27) Penalized linear models (ridge/lasso/elastic net)

Formulation. Ridge: $\min_\beta \|o - I\beta\|_2^2 + \lambda \|\beta\|_2^2$. Lasso: $+\lambda \|\beta\|_1$. EN: mix.
- **Where:** high-dimensional regression.
- **History:** ridge (1970), lasso (1996).

- **Use cases:** shrinkage, feature selection, stability.
- **Comprises:** penalty type, path algorithms (LARS), CV.

28) Reduced-rank regression (RRR)

Formulation. Multi-output: $O \approx IB$ with $\mathrm{rank}(B) \le r$ (factorization $B = UV^\top$).
- **Where:** multivariate outputs.
- **History:** 1950s–1970s multivariate stats.
- **Use cases:** dimension-reduced mappings, CCA-like tasks.
- **Comprises:** low-rank constraint; SVD-based solutions.

29) Mixture density network (MDN)

Formulation. NN outputs $\{\pi_m(i), \mu_m(i), \Sigma_m(i)\}$; $p(o \mid i) = \sum_m \pi_m \mathcal{N}(o; \mu_m, \Sigma_m)$. - **Where:** multimodal continuous targets.
- **History:** Bishop (1994).
- **Use cases:** inverse kinematics, ambiguous regressions.
- **Comprises:** neural backbone; mixture NLL training.

30) Conditional normalizing flow (cNF)

Formulation. Invertible $z = f_\theta(o; i)$; $\log p(o \mid i) = \log p_Z(z) + \log\left|\det \frac{\partial f_\theta}{\partial o}\right|$. - **Where:** flexible conditional densities.
- **History:** 2015–2018 (NICE/RealNVP/Glow).
- **Use cases:** generative regression, SBI, simulation surrogates.
- **Comprises:** invertible blocks, base density, exact likelihood.

31) Copula-based regression

Formulation. $F_{O,I}(o,i) = C(F_O(o), F_I(i))$; derive $p(o \mid i)$ via conditional copula.
- **Where:** dependence beyond correlation.
- **History:** Sklar (1959); applied widely in finance/insurance.
- **Use cases:** joint risk, tail dependence.
- **Comprises:** marginal models + copula family (Gaussian/t/Archimedean).

32) Cox proportional hazards (survival)

Formulation. $\lambda(t \mid i) = \lambda_0(t) \exp(\beta^\top i)$; partial likelihood for β . - **Where:** biostatistics, churn/time-to-event.
- **History:** Cox (1972).
- **Use cases:** retention, reliability.
- **Comprises:** baseline hazard, proportional effect, censoring handling.

33) Ordinal regression (proportional odds)

Formulation. $\Pr(o \le c \mid i) = \sigma(\theta_c - \beta^\top i)$ for ordered classes c . - **Where:** ratings, grades, stages.
- **History:** McCullagh (1980).
- **Use cases:** Likert outcomes, severity scales.
- **Comprises:** cutpoints θ_c , shared slope(s), logit/probit link.

34) Zero-inflated / hurdle count models

Formulation. Mixture with inflation at zero: $\Pr(o = 0 \mid i) = \pi(i) + [1 - \pi(i)]f(0 \mid i)$, else $o \sim f(\cdot \mid i)$ (Poisson/NB). - **Where:** sparse counts.
- **History:** 1990s.
- **Use cases:** claims, defects, clicks.
- **Comprises:** zero-process + count component; logit + log link.

35) Conditional random field (CRF)

Formulation. $p(o_{1:T} \mid i_{1:T}) \propto \exp\big(\sum_t \theta^\top f(o_{t-1}, o_t, i_{1:T}, t)\big)$. - **Where:** structured prediction.
- **History:** Lafferty et al. (2001).
- **Use cases:** NER, segmentation, labeling.
- **Comprises:** feature potentials, global normalization; DP inference.

36) Energy-based model (EBM)

Formulation. Define energy $E_\theta(o, i)$; $p(o \mid i) \propto e^{-E_\theta(o,i)}$. - **Where:** generative modeling, anomaly detection.
- **History:** Boltzmann machines (1980s); modern EBMs (2000s+).
- **Use cases:** denoising, score-based methods, retrieval.
- **Comprises:** energy network; contrastive/score training; sampling.

37) Volterra series

Formulation. $o_t = \sum_\tau h_1(\tau) i_{t-\tau} + \sum_{\tau_1,\tau_2} h_2(\tau_1, \tau_2) i_{t-\tau_1} i_{t-\tau_2} + \cdots$. - **Where:** weakly nonlinear systems.
- **History:** early 20th c.
- **Use cases:** RF, biomedical devices, loudspeaker modeling.
- **Comprises:** kernels of increasing order; truncation/regularization.

38) Hammerstein–Wiener block models

Formulation. Static nonlinearity → LTI (Hammerstein) or LTI → nonlinearity (Wiener): $o_t = (H * g(i))_t$ or $o_t = g\big((H * i)_t\big)$. - **Where:** control and system ID.
- **History:** 1930s–1950s.
- **Use cases:** actuator/sensor nonlinearities, saturation.
- **Comprises:** choice/order of blocks; identification per block.

39) NARX (nonlinear AR with exogenous input)

Formulation. $o_t = F(o_{t-1:t-p},\, i_{t:t-q}) + \epsilon_t$. - **Where:** nonlinear time-series with drivers.
- **History:** Billings (1980s).
- **Use cases:** industrial processes, macro, energy load.
- **Comprises:** lag selection, nonlinear F (e.g., NN), regularization.

40) Nonlinear SDE state-space

Formulation. $dx_t = f(x_t, i_t)\,dt + G\,dW_t$, measurement $o_t = h(x_t) + v_t$; discretize for inference. - **Where:** stochastic dynamics.
- **History:** Itô calculus (1940s); EKF/UKF/particle filters (1960s–1990s).
- **Use cases:** finance, biology, target tracking.
- **Comprises:** drift/diffusion, measurement model, Bayesian filters.

Quick selection guidance

- **Tabular, little preprocessing:** Trees/GBDT (7), GLM/GAM (21–22), penalized linear (27).
- **Uncertainty needed with small N:** GP (8), Bayesian (10).
- **Sequential with control:** Kalman/State-space (14), ARIMAX (15), IO-HMM (16), NARX (39).
- **Long sequences/text:** RNN/LSTM/GRU (17), Transformer (18).
- **Multimodal outputs:** MoE (9), MDN (29), cNF (30).
- **Causal questions:** SEM/DAGs (12).
- **Counts/zero-heavy:** GLM (Poisson/NB) (21), zero-inflated/hurdle (34).
- **Ordered labels / survival:** Ordinal (33), Cox (32).

Want this exported to PDF/Word or trimmed to a one-pager cheat sheet? I can generate that too.

---

41) Discrete Differential Geometry (DDG)

Formulation. Work with meshes (vertices/edges/faces) and discrete analogues of smooth operators (gradient, divergence, Laplace–Beltrami) so that key theorems (Stokes, Gauss–Bonnet) hold in the discrete. A common operator is the cotangent Laplacian on a triangle mesh: (Delta f)*i = (1/(2 A_i)) * sum* (cot alpha_ij + cot beta_ij) * (f_j - f_i). - **Where:** geometry processing, simulation on surfaces, mesh optimization. - **History:** 2000s–present as a program to discretize the theory (not just equations) of differential geometry. - **Use cases:** curvature/normal computation, mean-curvature flow, geodesics (heat method), discrete shells/cloth, discrete electromagnetics. - **Comprises:** mesh + discrete forms, Hodge stars, exterior derivative/codifferential, sparse matrices; ensures convergence/invariants under refinement.

42) Discrete Exterior Calculus (DEC)

Formulation. Extend calculus of differential forms to meshes/simplicial complexes. Discrete exterior derivative d maps k-forms to (k+1)-forms, Hodge star maps primal to dual, Laplace–de Rham Delta = delta d

+ d delta with delta = (Hodge)^{-1} d^T (Hodge). - **Where:** mesh-based PDEs, physics on curved domains, graphics. - **History:** 2000s formalization; widely adopted in graphics/geometry computing. - **Use cases:** Poisson/Helmholtz on surfaces/volumes, incompressible flow on manifolds, EM on irregular grids, vector-field design. - **Comprises:** simplicial complex, cochain spaces (0/1/2-forms), coboundary matrices, Hodge stars (metric), structure-preserving assembly.

43) Discrete Poisson Solver

Formulation. Solve Delta phi = rho (or Laplace's equation Delta phi = 0) on a grid/mesh using a discrete Laplacian L: assemble sparse L and solve L Phi = b (Dirichlet/Neumann/Robin boundary conditions). - **Where:** everywhere in graphics & simulation. - **History:** classic numerical analysis; cotan-Laplacian and multigrid make it scalable. - **Use cases:** pressure projection in fluids, Poisson image editing, surface reconstruction, harmonic maps, smoothing. - **Comprises:** stencil/cotan Laplacian, mass matrix (optional), boundary handling, direct or iterative solvers (CG/AMG/FFT on grids).

44) Discrete Laplace–Beltrami Operator

Formulation. Cotangent formula on triangle meshes (as above); generalized eigenproblem C x = lambda M x gives spectral bases. - **Where:** foundational operator for mesh processing. - **History:** cotan weights popularized in 1990s–2000s geometry processing. - **Use cases:** smoothing/diffusion, spectral analysis, parameterization, curvature/shape editing, heat method for geodesics. - **Comprises:** symmetric weight matrix C, lumped/consistent mass matrix M, sparsity and maximum-principle properties.

45) Discrete Hodge Decomposition

Formulation. For an edge-based 1-form/vector field xi on a mesh, decompose xi = d phi + delta psi + h (exact + coexact + harmonic), by solving Poisson-type systems for potentials. - **Where:** fluid projection, vector-field analysis/design, graph flow analysis. - **History:** discrete analog of Hodge theory; DEC enables practical algorithms. - **Use cases:** enforcing incompressibility, separating cycles from gradients on meshes/graphs, topology inference. - **Comprises:** incidence matrices, Hodge stars, linear solves for potentials; harmonic basis lives in cohomology nullspaces.

46) Discrete Conformal Mapping

Formulation. Angle-preserving maps via (i) discrete Ricci flow on vertex scale factors u solving Delta u = K_target − K(u), (ii) LSCM/ABF energies enforcing Cauchy–Riemann conditions, or (iii) circle packings. - **Where:** UV parameterization, uniformization, medical/architectural geometry. - **History:** discrete uniformization & Ricci flow (2000s+), LSCM/ABF in graphics. - **Use cases:** low-distortion texture maps, spherical/disk parameterizations, remeshing, surface comparison/registration. - **Comprises:** target curvatures/boundary constraints, Laplacian systems, nonlinear solves; outputs (u,v) coords or new metric (edge lengths).

47) 3D Gaussian Splatting (real-time radiance fields)

Formulation. Scene represented by anisotropic 3D Gaussians {(mu_k, Sigma_k, alpha_k, c_k)} with per-Gaussian color modeled by low-order spherical harmonics. Images rendered by visibility-aware differentiable rasterization ("splatting") of elliptical Gaussians to pixels; parameters optimized from posed multi-view images. - **Where:** neural rendering & graphics pipelines needing real-time novel-view synthesis (AR/VR, telepresence, robotics mapping). - **History:** Kerbl et al. (2023) as a fast alternative to NeRF MLP volume rendering; builds on point/volume splatting (Zwicker et al., 2001/2002). - **Use cases:** photorealistic view synthesis, rapid scene capture/playback, interactive 3D editing, real-time teleoperation/SLAM visualization; practical complement to NeRFs. - **Comprises:** anisotropic 3D Gaussians with opacity/color (spherical harmonics), alternating optimization of positions/scales/opacities, visibility culling, tile-based GPU renderer (approx. 1080p at or above 30 fps); official implementations exist.

48) Elliptic PDEs — Laplace's Equation

Formulation. Delta u = 0 on a domain with boundary conditions (Dirichlet/Neumann/Robin). Minimizes Dirichlet energy integral of |grad u|^2. - **Where:** steady-state potential/heat problems, geometry processing, electrostatics. - **History:** classical (Laplace); cornerstone of potential theory. - **Use cases:** harmonic interpolation/parameterization, smoothing, electrostatic/gravitational potential without sources, incompressible flow stream-functions. - **Comprises:** PDE + BCs; discretization via FEM/FDM/FVM; linear sparse solves, maximum principle.

49) Elliptic PDEs — Poisson's Equation

Formulation. Delta u = f with source term f and boundary conditions; generalizes Laplace. - **Where:** steady diffusion with sources/sinks; ubiquitous in simulation/graphics. - **History:** Poisson (1812); numerical Poisson solvers are core linear-algebra workloads. - **Use cases:** pressure projection in fluids, Poisson image editing, reconstruction from normals/divergence, electrostatics, gravity. - **Comprises:** discrete Laplacian assembly, right-hand side from sources/BCs, direct/iterative/multigrid/FFT solvers.

50) Ordinary Differential Equations (ODEs)

Formulation. x_dot = f(x(t), i(t), theta); o(t) = h(x(t)). - **Where:** physical/biological/chemical dynamics, macro models. - **History:** classical calculus; numerical IVP solvers (Runge–Kutta, BDF). - **Use cases:** simulation, forecasting with mechanistic priors, digital twins. - **Comprises:** state, vector field, initial conditions; integrators, sensitivity.

51) Linear Time-Varying ODEs (LTV)

Formulation. x_dot = A(t) x + B(t) i; o = C(t) x + D(t) i. - **Where:** control with scheduled parameters, aerospace, time-varying systems. - **History:** 20th-century control theory; Floquet/LTV analysis. - **Use cases:** gain-scheduling, time-varying filters, tracking. - **Comprises:** superposition holds; fundamental matrix, time-varying LQR.

52) Differential–Algebraic Equations (DAEs)

Formulation. F(x_dot, x, i, t, theta) = 0 with algebraic constraints. - **Where:** power systems, multibody mechanics, chemical processes. - **History:** DAE index theory, specialized solvers (IDA/DAE solvers). - **Use cases:** constrained dynamics, circuit simulation (SPICE), process models. - **Comprises:** index (0/1/2), consistent initialization, implicit integrators.

53) Delay Differential Equations (DDEs)

Formulation. x_dot(t) = f(x(t), x(t−tau), i(t), theta). - **Where:** population dynamics, control with transport/ latency. - **History:** renewal equations; numerical methods with history buffers. - **Use cases:** epidemiology with incubation, networked control. - **Comprises:** history function as extended state; stability vs. delay.

54) Piecewise-Smooth / Hybrid ODEs

Formulation. Mode-dependent x_dot = f_m(...) with guards/resets for switches/impacts. - **Where:** power electronics, contact dynamics, on/off control. - **History:** hybrid systems theory (1990s+). - **Use cases:** thermostats, DC-DC converters, legged robots. - **Comprises:** automaton (modes/guards/resets), event detection, complementarity.

55) Partial Differential Equations (PDEs)

Formulation. u_t = L[u; i(x,t), theta], outputs as fields or functionals (e.g., o = integral h(u) dx or boundary traces). - **Where:** heat/flow/waves, diffusion–reaction, option pricing. - **History:** classical continuum mechanics and analysis. - **Use cases:** thermal/structural/CFD, image processing, geophysics. - **Comprises:** spatial discretization (FEM/FDM/FVM), time stepping, BCs/ICs.

56) Stochastic Differential Equations (SDEs)

Formulation. dx = f(x,i) dt + G(x,i) dW_t; o = h(x) + v. - **Where:** finance, biology, tracking. - **History:** Ito/ Stratonovich calculus; numerical SDE schemes. - **Use cases:** noisy dynamics, volatility models, stochastic control. - **Comprises:** drift/diffusion terms, discretization (Euler–Maruyama, Milstein), filtering.

57) Controlled State-Space Models (nonlinear)

Formulation. x_dot = f(x,i); o = h(x,i). - **Where:** robotics, avionics, process control (LQR/MPC around linearizations). - **History:** modern control; nonlinear observers/filters. - **Use cases:** tracking/estimation, regulation, trajectory following. - **Comprises:** dynamics + outputs, observers (EKF/UKF), MPC.

58) Optimal Control with ODE Constraints

Formulation. Minimize J = integral l(x,i) dt subject to x_dot = f(x,i), o = h(x). - **Where:** trajectory planning, dosing schedules, energy systems. - **History:** Pontryagin’s maximum principle; direct collocation/shooting. -

Use cases: robotics trajectories, drug dosing, EV routing. - **Comprises:** adjoints/necessary conditions, NLP transcriptions, constraints.

59) Neural ODEs (and controlled neural ODEs)

Formulation. x_dot = f_theta(x,i,t) with neural-net vector field; o = h_phi(x); train via adjoint/sensitivity. - **Where:** learned dynamics, irregularly sampled time series. - **History:** 2018-; extensions with control/latent ODEs. - **Use cases:** system ID, continuous-time sequence models, generative flows. - **Comprises:** differentiable solvers, stability/regularization (Lipschitz, spectral norms).

60) Physics-Informed Neural Networks (PINNs)

Formulation. Train neural net u_theta to satisfy PDE/ODE residuals + data by minimizing L = || N[u_theta] − f ||^2 + BC/IC losses + data loss. - **Where:** PDE-constrained learning/inverse problems with sparse data. - **History:** 2017-; rapidly developing variants (domain decomposition, hard constraints). - **Use cases:** parameter inference, surrogate modeling, closure discovery. - **Comprises:** automatic differentiation of residuals, collocation points, normalization/weights.

61) Operator Learning (DeepONet / FNO)

Formulation. Learn an operator mapping i(·) -> o(·). DeepONet uses branch/trunk nets; FNO learns in Fourier space with spectral convolutions. - **Where:** fast PDE surrogates, real-time control/optimization. - **History:** 2019-2021+; rapid adoption in scientific ML. - **Use cases:** weather/ocean surrogates, CFD, inverse problems. - **Comprises:** function-space inputs/outputs, spectral layers, training on ensembles of fields.

62) Hammerstein/Wiener with ODE cores

Formulation. Static nonlinearity g(·) composed with LTI/LTV ODE system (or reverse): input shaper -> dynamics -> output nonlinearity. - **Where:** actuators/sensors with saturation & dynamics; system ID. - **History:** classical block-structured models; modern gray-box ID. - **Use cases:** servo/drive systems, mechatronics. - **Comprises:** parametric blocks (nonlinearities, ODEs), identification via least squares + nonlinear optimization.

63) Compartmental ODEs (SIR, kinetics)

Formulation. Mass-action couplings among compartments; x_dot = S r(x); o = h(x). - **Where:** epidemiology, pharmacokinetics/pharmacodynamics, chemical reactions. - **History:** 1920s-; standard in life sciences. - **Use cases:** disease spread, dosing/absorption, reaction networks. - **Comprises:** stoichiometry matrix S, rate laws/parameters; identifiability from time series.

64) Hawkes with ODE Embedding

Formulation. Self-/mutually-exciting point processes whose intensity dynamics are represented by an auxiliary ODE/state z approximating kernel memory; events update z, and lambda(t) = g(z(t)). - **Where:** high-frequency events (finance, seismicity, social), forecasting spikes. - **History:** Hawkes (1971); recent neural/ODE embeddings for tractable learning. - **Use cases:** event prediction, anomaly/spike detection, influence modeling. - **Comprises:** event history, kernel/ODE surrogate, thinning/likelihood training; links point processes with continuous states.